Sequential Scalar Quantization of Two Dimensional Vectors in Polar and Cartesian Coordinates

SEQUENTIAL SCALAR QUANTIZATION OF TWO DIMENSIONAL VECTORS IN POLAR AND CARTESIAN COORDINATES

BY

HUIHUI WU, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2018) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Sequential Scalar Quantization of Two Dimensional Vectors in Polar and Cartesian Coordinates
AUTHOR:	Huihui Wu M.Sc., (Communication Engineering) Xiamen University, Xiamen, China
SUPERVISOR:	Dr. Sorina Dumitrescu
NUMBER OF PAGES:	xv, 146

To my beloved family

Abstract

This thesis addresses the design of quantizers for two-dimensional vectors, where the scalar components are quantized sequentially. Specifically, design algorithms for unrestricted polar quantizers (UPQ) and successively refinable UPQs (SRUPQ) for vectors in polar coordinates are proposed. Additionally, algorithms for the design of sequential scalar quantizers (SSQ) for vectors with correlated components in Cartesian coordinates are devised. Both the entropy-constrained (EC) and fixed-rate (FR) cases are investigated.

The proposed UPQ and SRUPQ design algorithms are developed for continuous bivariate sources with circularly symmetric densities. They are globally optimal for the class of UPQs/SRUPQs with magnitude thresholds confined to a finite set. The time complexity for the UPQ design is $O(K^2 + KP_{max})$ in the EC case, respectively $O(KN^2)$ in the FR case, where K is the size of the set from which the magnitude thresholds are selected, P_{max} is an upper bound for the number of phase levels corresponding to a magnitude bin, and N is the total number of quantization bins. The time complexity of the SRUPQ design is $O(K^3P_{max})$ in the EC case, respectively $O(K^2N'^2P_{max})$ in the FR case, where N' denotes the ratio between the number of bins of the fine UPQ and the coarse UPQ.

The SSQ design is considered for finite-alphabet correlated sources. The proposed algorithms are globally optimal for the class of SSQs with convex cells, i.e, where each

quantizer cell is the intersection of the source alphabet with an interval of the real line. The time complexity for both EC and FR cases amounts to $O(K_1^2 K_2^2)$, where K_1 and K_2 are the respective sizes of the two source alphabets. It is also proved that, by applying the proposed SSQ algorithms to finite, uniform discretizations of correlated sources with continuous joint probability density function, the performance approaches that of the optimal SSQs with convex cells for the original sources as the accuracy of the discretization increases.

The proposed algorithms generally rely on solving the minimum-weight path (MWP) problem in the EC case, respectively the length-constrained MWP problem or a related problem in the FR case, in a weighted directed acyclic graph (WDAG) specific to each problem. Additional computations are needed in order to evaluate the edge weights in this WDAG. In particular, in the EC-SRUPQ case, this additional work includes solving the MWP problem between multiple node pairs in some other WDAG. In the EC-SSQ (respectively, FR-SSQ) case, the additional computations consist of solving the MWP (respectively, length-constrained MWP) problem for a series of other WDAGs.

Acknowledgements

I would like to express the heartfelt gratitude to my supervisor, Dr. Sorina Dumitrescu. Dr. Dumitrescu is a patient and nice supervisor who inspires me to conduct research, and I have benefited a lot from her rigorous attitude to scientific research, and the enthusiasm to brace challenges. I cannot thank her enough for advising me during all these four years.

I am also very grateful to my committee members, Dr. Jun Chen and Dr. Shahram Shirani, for their insightful instructions and suggestions on the past committee meetings and on the thesis. Also I want to express my sincere appreciation to the external examiner for the valuable comments to improve the thesis. Moreover, many thanks to Mrs. Cheryl Gies for helping coordinate the defence related matters.

I am thankful to my Master's thesis supervisor, Dr. Lin Wang, and to the former group members who have been accompanying me since then.

I would also like to thank all my friends who helped me all the way to the final defence, especially those in the multimedia lab.

At last, I am truly grateful to my parents for their support and love. Most importantly, words are powerless to express my gratitude to my wonderful wife, and I appreciate what she has done more than she knows.

List of Abbreviations

EC: Entropy-constrained

ECRQ: Entropy-constrained rectangular quantizer

ECSPQ: Entropy-constrained strictly polar quantizer

ECUPQ: Entropy-constrained unrestricted polar quantizer

ECVQ: Entropy-constrained vector quantizer

FR: Fixed-rate

FRUPQ: Fixed-rate unrestricted polar quantizer

MWP: Minimum-weight path

pdf: Probability density function

pmf: Probability mass function

SCCS: Sequential coding of correlated sources

SPQ: Strictly polar quantizer

SR: Successively refinable

SRSPQ: Successively refinable strictly polar quantizer

SRSQ: Successively refinable scalar quantizers

SRUPQ: Successively refinable unrestricted polar quantizer

SSQ: Sequential scalar quantizer

UPQ: Unrestricted polar quantizer

WDAG: Weighted directed acyclic graph

Notations

$\mathbb{R}$: Real space

$|\ \mathcal{X}\ |$: Cardinality of a finite set $\mathcal{X}$

$\mathbb{Z}_+$: Set of positive integers

$\mathcal{X} \setminus \mathcal{Y}$: A set containing elements in $\mathcal{X}$ but not in $\mathcal{Y}$

$\lceil \cdot \rceil$: Ceiling function

$\lfloor \cdot \rfloor$: Floor function

$\lfloor \cdot \rceil$: Rounding function

$\| \cdot \|_m$: m-norm

$\mathbb{P}[\cdot]$: Probability of an event

$\mathbb{E}[\cdot]$: Expectation function

List of Tables

Contents

List of Figures

Chapter 1

Introduction

Quantization has been widely utilized in analog-to-digital conversion and data compression in modern telecommunications. Quantizers can be divided into two classes (Gray and Neuhoff, 1998): scalar quantizers, which operate on individual samples, and vector quantizers, which operate on groups of samples. Despite the fact that non-structured vector quantizers with large dimension outperform the scalar quantizers, their computational and storage complexity grows exponentially with the dimension. In addition, it is worth pointing out that the rate gap between the optimum entropy-constrained scalar quantizer and the rate-distortion limit (achieved using vector quantization with infinite dimension) is only 0.2546 bits/sample (Gish and Pierce, 1968) at high resolution for memoryless sources. Therefore, scalar quantizers are commonly used, especially in image and video compression algorithms, including the well known JPEG standard (Wallace, 1992), JPEG 2000 standard (Skodras *et al.*, 2001; Taubman and Marcellin, 2012) and H.264/AVC standard (Wiegand *et al.*, 2003).

However, for sources with memory, independent quantization of each scalar component is too wasteful. A technique to overcome this drawback is the application of a transform on blocks of samples in order to decorrelate the signal, followed by scalar

quantization of transform coefficients. Alternatively, sequential scalar quantization has also been investigated (Gray and Neuhoff, 1998; Balasubramanian *et al.*, 1995). A sequential scalar quantizer (SSQ) quantizes each scalar component of a vector sequentially, the quantizer of each component depending on the previous components. This technique outperforms the use of independent scalar quantizers for each component since it is able to exploit the correlation between components. The SSQ is also of interest in situations where the encoder has sequential access to the source samples, such as in sequential coding of frames of a video sequence.

In the case of two-dimensional sources with circularly symmetric probability densities, the approach of quantizing each scalar component of the vector represented in polar coordinates was also considered. This technique is termed polar quantization. More specifically, a polar quantizer consists of a quantizer for the magnitude followed by a uniform quantizer for the phase. A strictly polar quantizer (SPQ) uses independent quantizers for the magnitude and phase (Pearlman, 1979). In unrestricted polar quantization the phase quantizer depends on the magnitude level (Wilson, 1980). The unrestricted polar quantizers (UPQ) were shown to outperform their strict counterparts. Notice that the UPQ can also be regarded as a sequential scalar quantizer applied to the polar coordinates of a vector. Polar quantization is useful in numerous applications, such as image processing (Senge, 1977; Kingsbury and Reeves, 2003), for the encoding of discrete Fourier transform coefficients (Gallagher, 1978; Pearlman and Gray, 1978), in holographic image processing (Bruckstein *et al.*, 1998), as well as for the quantization of sinusoid signals with application in audio coding (Vafin and Kleijn, 2005). More recently, polar quantization was also used for wireless receiver design in (Nazari *et al.*, 2014).

This thesis addresses the design of SSQs for two dimensional correlated sources, and of UPQs and successively refinable UPQs (SRUPQ) for bivariate sources with

circularly symmetric densities. Prior design algorithms for UPQs, SRUPQs and SSQs are either based on high-rate quantization theory or suffer from other drawbacks, which will be discussed in detail in the next section. This thesis proposes efficient design algorithms with guaranteed optimality properties under certain conditions.

The rest of the chapter is organized as follows. Section 1.1 reviews the related literature and presents the motivation for our work. Section 1.2 describes the contribution of this thesis. The organization of this thesis and the list of publications resulted from the related research work are given in Section 1.3.

1.1 Motivation and Related Work

1.1.1 Unrestricted Polar Quantizers

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 non-uniform. Polar quantization of bivariate sources with circularly symmetric densities, has been extensively investigated either for the general case or for the specific Gaussian case, e.g., see references (Senge, 1977; Gallagher, 1978; Pearlman and Gray, 1978; Pearlman, 1979; Bucklew and Gallagher, 1979a,b; Wilson, 1980; Swaszek and Thomas, 1982; Swaszek, 1985; Swaszek and Ku, 1986; Neuhoff, 1997; Moo and Neuhoff, 1998; Peric and Stefanovic, 2002; Vafin and Kleijn, 2005; Petković *et al.*, 2011; Perić and Nikolić, 2013; Jovanović *et al.*, 2016; Pobloth *et al.*, 2005; Ravelli and Daudet, 2007).

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 (Swaszek, 1985; Moo and Neuhoff, 1998; Jovanović *et al.*, 2016) for the strict case and in (Peric and Stefanovic, 2002) for the unrestricted case. We point out that the above mentioned papers assume fixed-rate (FR) quantization, i.e., where the goal is to minimize the distortion for a fixed number of total quantization levels. The asymptotic analysis of FR non-uniform UPQ was addressed in (Swaszek and Ku, 1986; Neuhoff, 1997; Perić and Nikolić, 2013). Additionally, note that such techniques guarantee the optimality of the design only as the number of quantization levels approaches infinity.

The design of optimal practical polar quantizers, i.e., without the high rate assumption, was considered in (Gallagher, 1978; Pearlman, 1979) for the FRSPQ and in (Wilson, 1980) for the FRUPQ. 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 (Pearlman, 1979; Wilson, 1980) by using an iterative algorithm similar to Max-Lloyd algorithm (Max, 1960; Lloyd, 1982) 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 (Pearlman,

1979; Wilson, 1980) rely on exhaustive search to optimize the rate allocation.

The authors of (Petković *et al.*, 2011) propose a nearly optimal algorithm for FRUPQ, which iteratively optimizes the values of the vector of decision thresholds, respectively, $(P_1, \cdots, P_M)$ and the vector of reconstruction levels, while the other two vectors are kept fixed. The drawbacks of the method in (Petković *et al.*, 2011) are slow convergence and lack of guarantee of optimality.

Therefore, the above discussion motivates our search for a tractable and globally optimal design algorithm for FRUPQ, for finite rates.

Further, in order to increase the efficiency of the polar quantizer, entropy coding may be applied to the quantizer's outputs. This was done, for instance, in (Wilson, 1980). However, for optimal performance the polar quantizer has to be optimized under a constraint on the entropy. Such a quantizer is called entropy-constrained (EC) quantizer. Work (Vafin and Kleijn, 2005) is the only work addressing the design of EC polar quantizers, up to our knowledge. The authors of (Vafin and Kleijn, 2005) derive the asymptotically optimal ECUPQ and 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 EC 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 (Vafin and Kleijn, 2005) 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 (Vafin and Kleijn, 2005) 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 (Vafin and Kleijn, 2005) 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 (Vafin and Kleijn, 2005) holds as the rate approaches infinity, but it is not guaranteed at finite rates.

The above observations raise two natural questions that motivate the practical design of ECUPQ:

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.

1.1.2 Successively Refinable Unrestricted Polar Quantizers

A successively refinable (SR) quantizer encodes the source into a sequence of embedded bitstreams, which enables the decoder to retrieve the source reconstruction in a progressively refinable manner. Specifically, a coarse reconstruction can be obtained by decoding the base layer, while the quality of the reconstruction improves as more refinement layers are decoded. As a promising technique for broadcasting multimedia to heterogeneous devices over fluctuating bandwidth or unreliable networks, research

topics related to SR quantization have drawn significant attention, see (Equitz and Cover, 1991; Rimoldi, 1994; Brunk and Farvardin, 1996; Jafarkhani and Tarokh, 1999; Dumitrescu and Wu, 2004; Effros and Dugatkin, 2004; Chen *et al.*, 2010; Wang and Gastpar, 2014; No *et al.*, 2016; Kostina and Tuncel, 2017). Notably, the simplified bit plane coding variant of the SR quantizer has been adopted as the baseline quantization method of the JPEG 2000 image compression standard (Skodras *et al.*, 2001; Taubman and Marcellin, 2012).

Consequently, it is natural to raise the curiosity on the design of successively refinable UPQ (SRUPQ).

Work (Ravelli and Daudet, 2007) is the only work addressing the design of SR polar quantizers, up to our knowledge. The authors of (Ravelli and Daudet, 2007) consider the fixed-rate designs of SRUPQ and successively refinable strictly polar quantizer (SRSPQ). In the latter case, both the low-rate and high-rate quantizers are designed, where either the number of magnitude bins or the number of phase regions is doubled (for high-rate case), or halved (for low-rate quantizer). In the case of fixed-rate SRUPQ (FR-SRUPQ), the authors consider only the high-rate quantizers, and an analytical solution using asymptotic quantization theory is derived, to determine whether a magnitude refinement or a phase refinement should be applied (i.e., the one gives smaller distortion).

This thesis concerns the practical performance at finite rates, for both the EC-SRUPQ and FR-SRUPQ designs.

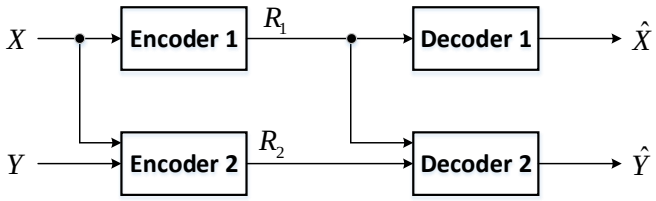


Figure 1.1: Block diagram of a sequential code for correlated sources.

1.1.3 Scalar Quantizer for Sequential Coding of Correlated Sources

The problem of sequential coding of correlated sources (SCCS) in an information theoretical sense was introduced in (Viswanathan and Berger, 2000). The authors of (Viswanathan and Berger, 2000) gave a complete characterization of the achievable rate-distortion region.

Figure 1.1 illustrates the framework of SCCS, where (X, Y) is a pair of jointly distributed random variables. First encoder 1 observes only the source X and encodes it at rate R_1. Decoder 1 receives the output of encoder 1 and reconstructs an estimation $\hat{X}$ of X. Subsequently, encoder 2 observes both X and Y and generates a description of Y at rate R_2. Decoder 2 then utilizes the outputs of both encoders to reconstruct an estimation $\hat{Y}$ of source Y. Note that the problem of SCCS can be regarded as a generalization of the successive refinement coding problem (Rimoldi, 1994) since it reduces to the latter when the two sources are equal. In practice, the SCCS problem can be utilized to model a video source, where a sequence of frames corresponds to a sequence of correlated sources. Moreover, it also provides a theoretical model for video compression using frame-differencing, as the encoding of a later frame refers to a previous frame.

In this thesis, we address the problem of designing a practical coding scheme for the SCCS problem, which uses scalar quantization at each encoder. Specifically, encoder

1 consists of a unique scalar quantizer for the source X, while encoder 2 consists of a set of scalar quantizers for the source Y, each quantizer corresponding to a particular output of encoder 1. We refer to such a scheme using the term sequential scalar quantizer (SSQ). Past work on the design of SSQ includes (Balasubramanian *et al.*, 1995) and (Chang and Allebach, 1993), where only the FR case is considered and the quantizers are derived based on the asymptotic quantization theory. Specifically, the authors of (Balasubramanian *et al.*, 1995) find closed form expressions for the distortion resulting from SSQ as a function of the quantizer design parameters and find the optimum parameter values that minimize the distortion. The proposed SSQ technique is utilized for color palette design of RGB images in (Balasubramanian *et al.*, 1994), whereas an initial SSQ structure has to be preset in order to obtain the optimal number of quantization levels. It is worth pointing out that the optimization in (Balasubramanian *et al.*, 1995) is greedy. Further, the authors of (Chang and Allebach, 1993) improve the performance of the design procedure by considering the distribution of the unquantized scalars as well.

This thesis addresses the problem of optimal SSQ design for finite-alphabet sources in both the FR and EC cases. Note that past work (Balasubramanian *et al.*, 1995; Chang and Allebach, 1993) did not consider EC-SSQ, while the optimality claims for the FR-SSQ design algorithms hold only asymptotically as the rate approaches infinity.

1.2 Contribution

This section provides the detailed contribution of this thesis.

1.2.1 Design of Unrestricted Polar Quantizer

In this thesis we propose efficient globally optimal UPQ design algorithms, for both EC and FR cases, for the class of UPQs with magnitude quantizer thresholds restricted to a predefined finite set. In practice this finite set can be a fine uniform discretization of the interval $[0, B]$ for some sufficiently large B.

In the ECUPQ case, we formulate the optimization problem as the minimization of the Lagrangian for a given multiplier λ, which is the same formulation as in (Vafin and Kleijn, 2005). 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 in Section 1.1.1. 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 (Wilson, 1980) and the practical ECUPQ of (Vafin and Kleijn, 2005) designed based on the high rate assumption. The gain of our scheme over the scheme of (Wilson, 1980) 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. 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 (Vafin and Kleijn, 2005). We have also compared the proposed ECUPQ with the ECRQ obtained using the algorithm of (Muresan and Effros, 2008) 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 JPEG 2000 image compression standard was the need to improve the performance at low bit rates (Skodras *et al.*, 2001; Taubman and Marcellin, 2012).

In the FRUPQ case, the solution algorithm is based on dynamic programming sped up with the aid of a fast matrix search technique in totally monotone matrices (Aggarwal *et al.*, 1987), and achieves the time complexity of $O(KN^2)$, where N is the total number of quantization bins.

It is worth pointing out that the design of FRUPQ has significant differences

versus the design of ECUPQ. Specifically, the design of FRUPQ minimizes the distortion with a constraint on the number of levels, while the problem of ECUPQ design is formulated as the unconstrained minimization of a weighted sum of distortion and entropy. These different formulations call for different solution approaches, with distinct time complexities. Additionally, in the FRUPQ case, we solve the problem for any possible number N of quantizer levels, while the algorithm for ECUPQ can find only the ECUPQs corresponding to points on the lower boundary of the convex hull of the set of entropy-distortion pairs.

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 (Aggarwal *et al.*, 1994), entropy-constrained quantizers (Muresan and Effros, 2002, 2008), Wyner-Ziv quantizers (Muresan and Effros, 2002, 2008), multi-resolution and multiple description quantizers (Dumitrescu and Wu, 2002, 2005, 2007; Muresan and Effros, 2002, 2008), as well as joint source-channel quantizer with random index assignment (Dumitrescu, 2016). The aspect which distinguishes the most the proposed ECUPQ and FRUPQ designs 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 this work resides in proposing the first algorithm which handles efficiently the problem of rate allocation between the magnitude and phase quantizers, while still guaranteeing the globally optimal solution (under certain constraints) at finite rates.

1.2.2 Design of Successively Refinable Unrestricted Polar Quantizer

This thesis also presents the design of SRUPQs with two refinement stages, for both EC and FR cases. The proposed algorithms are globally optimal under the constraint that the magnitude quantizers' thresholds are confined to finite sets, which are fine uniform discretizations of the same interval $[0, B]$ for some sufficiently large B.

The optimization problem for the EC-SRUPQ case is formulated as the minimization of a weighted sum of the distortions and entropies of the coarse and fine component UPQs. This formulation further enables the approach of converting the cost function (after evaluating the refined UPQ and the optimal phase quantizers corresponding to each coarse magnitude bin) to a summation of the costs of individual magnitude bins of the coarse UPQ. Therefore, this problem can be modeled as an MWP problem in a certain WDAG, where each edge represents a possible bin of the coarse magnitude quantizer. To achieve this goal, the proposed solution proceeds in a series of steps including solving the MWP problems for multiple node pairs in another WDAG, which corresponds to the refined UPQ. Further, the efficient algorithm of evaluating the optimal number of phase levels for all possible magnitude bins of the refined UPQ is also presented. The overall running time of the solution to EC-SRUPQ case is $O(K_1 K_2^2 P_{cmax})$, where K_1 and K_2 are the sizes of the sets of possible magnitude thresholds of the coarse UPQ and fine UPQ components, respectively, and P_{cmax} is the maximum number of phase levels of the coarse UPQ.

Another contribution of this thesis lies in the optimal design of FR-SRUPQ. The

proposed solution algorithm involves solving a series of dynamic programming problems, where each problem deals with a single description FRUPQ design. Additionally, we point out that the dynamic programming formulation allows the design of FR-SRUPQ with any possible number of quantizer levels. The overall time complexity of the proposed FR-SRUPQ design algorithm amounts to $O(K_1 K_2 N'^2 N_1)$, where N_1 is the number of bins of the coarse UPQ, while N' denotes the ratio between the number of bins of the fine UPQ and the coarse UPQ.

The algorithm for ECUPQ design relies on solving a single MWP problem in a certain WDAG in conjunction with a procedure to compute the edge weights. The proposed algorithm for optimal EC-SRUPQ design is much more involved and needs to solve the MWP problem for multiple node pairs, where the framework for ECUPQ can no longer be applied. As a result, it also has a higher time complexity than the algorithm of ECUPQ design. Similar observations can be made for the FR-SRUPQ case. The algorithm for FRUPQ design solves a single stage dynamic programming problem, which is now one basic step in the FR-SRUPQ framework, and the overall asymptotic running time also increases.

It is also important to discuss the relation between this work and the work on the design of successively refinable scalar quantizers (SRSQ) (Dumitrescu and Wu, 2004; Muresan and Effros, 2008; Dumitrescu and Wu, 2002). The SRSQ design algorithms in the aforementioned work also include steps resembling solving the MWP problem for multiple node pairs in a WDAG. The connection/similarity with the SRSQ stems from the existence of the embedded partitions of the magnitude quantizers in the SRUPQ. On the other hand, as the SRUPQ is essentially a two-dimensional quantizer, the need to optimize the phase quantizer for each magnitude bin adds an additional level of complexity to the SRUPQ design problem. More specifically, it makes the computation of the edge weights more involved than in the SRSQ case.

1.2.3 Design of Scalar Quantizer for Sequential Coding of Correlated Sources

The design of SSQs for finite-alphabet correlated sources in the FR and EC cases, is another contribution of this thesis. The proposed solutions are globally optimal for the class of EC-SSQs, respectively FR-SSQs with convex cells.

In the EC case we formulate the optimization problem as the minimization of a weighted sum of the distortions at the two decoders and the rates at the two encoders. Note that this formulation corresponds to determining the points on the lower boundary of the convex hull of the set of all quadruples of rates and distortions achievable using EC-SSQ. The proposed algorithm relies on solving the MWP problem in a series of appropriately constructed WDAGs. The time complexity of our solution amounts to $O(K_1^2 K_2^2)$, where K_1 and K_2 are the respective sizes of the two source alphabets.

In the FR case we impose the rate constraint at encoder 1 by fixing the number of levels of the corresponding quantizer. The rate of each quantizer at the second encoder equals the logarithm of its number of levels. As in (Balasubramanian *et al.*, 1995; Chang and Allebach, 1993) we allow different encoder 2 quantizers to have different numbers of cells and compute the encoder 2 rate as the expectation of the rates of component quantizers. The optimization problem is formulated as the minimization of a weighted sum of the distortions at the two decoders and of the rate at encoder 2. The main difference between the optimization problems in the EC and FR cases stems from the fact that in the EC case the rate of a quantizer can be written as a sum of rates corresponding to individual quantizer cells, which is not possible in the FR case. Because of this difference the solution to the FR problem is more involved. In particular, it needs to solve length-constrained MWP problems in a series of WDAGs, rather than unconstrained MWP problems as in the EC case. Using the

straightforward solution algorithm for the length-constrained MWP problems leads to a total time complexity of $O(K_1^2 K_2^3)$. We further show that in some of these WDAGs the edge weights satisfy the Monge property, fact which enables the speed up of the solution by a factor of K_2.

As mentioned earlier, in both the EC and FR cases we design the SSQ under the constraint of cell convexity. It is important to highlight that this constraint does not preclude the optimality of the quantizers for the source Y (see Figure 1.1) since the design of each such quantizer reduces to the problem of optimal scalar quantizer design for the conditional probability mass function (pmf) of Y given the particular output of the quantizer for X.

We point out that, in the case of continuous-alphabet sources, it is intuitive that approximate solutions to the EC-SSQ, respectively FR-SSQ, design problem can be obtained by applying the proposed algorithm to discretizations of the original sources. Another notable contribution of this thesis is a theoretical proof of the fact that the SSQ obtained in this way approaches the performance of the optimal SSQ (with convex cells) for the original sources as the discretization increases in accuracy, if the sources have a continuous joint probability density function (pdf).

Note that for the SRSQ design problem (Muresan and Effros, 2008; Dumitrescu and Wu, 2002, 2004), the goal is also to design a quantizer for the first encoder and conditional quantizers for the second encoder. The main difference is that for the SRSQ design problem all quantizers are designed for the same source. In SSQ scenario the quantizers operating at the different decoders are for distinct sources. It turns out that this generalization significantly complicates the problem, which can no longer be solved by simply extending the framework in (Dumitrescu and Wu, 2004; Muresan and Effros, 2008; Dumitrescu and Wu, 2002). To illustrate this point note that the proposed design algorithm runs in $O(K^4)$ time for both EC-SSQ and FR-SSQ, when

$K_1 = K_2 = K$. On the other hand, the optimal design of SRSQ can be performed in $O(K^3)$ time for the EC case (Muresan and Effros, 2008; Dumitrescu and Wu, 2002), respectively in $O(K^2)$ time for the FR case (Dumitrescu and Wu, 2004).

1.3 Thesis Layout and Related Publications

The rest of this thesis is structured as follows. Chapter 2 presents the algorithm for the design of UPQs, while the design of SRUPQs is treated in Chapter 3. Chapter 4 proposes the design of SSQs, and both EC and FR cases are considered in each aforementioned chapter. Chapter 5 finally concludes this thesis and proposes some ideas to be investigated in future work.

This thesis consists of results of original research conducted by myself, except for contributions made by my supervisor, Dr. Sorina Dumitrescu. The following is a list of the publications which resulted from this research.

The content of Chapter 2 has been published in

- Wu, H. and Dumitrescu, S. (2018). Design of optimal entropy-constrained unrestricted polar quantizer for bivariate circularly symmetric sources. In *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* IEEE.

- Wu, H. and Dumitrescu, S. (2018). Design of optimal entropy-constrained unrestricted polar quantizer for bivariate circularly symmetric sources. *IEEE Transactions on Communications,* **65**(5), 2169–2180.

- Wu, H. and Dumitrescu, S. (2018). Design of optimal fixed-rate unrestricted polar quantizer for bivariate circularly symmetric sources. *IEEE Signal Processing Letters,* **25**(5), 715–719.

The content of Chapter 3 is presented in

- Wu, H. and Dumitrescu, S. (2018). Design of optimal entropy-constrained successively refinable unrestricted polar quantizer for bivariate circularly symmetric sources. In *Proc. 2018 29th Biennial Symposium on Communications (BSC),* IEEE.

- Wu, H. and Dumitrescu, S. (2018). Design of successively refinable unrestricted polar quantizer, *to be submitted.*

The content of Chapter 4 is presented in

- Wu, H. and Dumitrescu, S. (2017). Design of optimal entropy-constrained scalar quantizer for sequential coding of correlated sources. In *Proc. 2017 IEEE Information Theory Workshop (ITW),* pages 524–528. IEEE.

- Wu, H. and Dumitrescu, S. (2018). Design of optimal scalar quantizer for sequential coding of correlated sources, *IEEE Transactions on Communications*, A revision was submitted.

Chapter 2

Design of Unrestricted Polar Quantizer for Bivariate Circularly Symmetric Sources

This chapter proposes algorithms for the design of UPQs for bivariate circularly symmetric sources, for both EC and FR cases. The algorithms are globally optimal for the class of UPQs with magnitude quantizers' thresholds confined to a finite set.

The optimization problem for ECUPQ 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 MWP problem in a certain WDAG. 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 solution algorithm of FRUPQ is based on dynamic programming, which is further accelerated by exploiting a monotonicity property of the cost function. The time

complexity of the accelerated algorithm is $O(KN^2)$, where N is the number of target qunatizer levels.

The practical performance of the proposed algorithms is assessed for a bivariate circularly symmetric Gaussian source. Our results demonstrate that the proposed ECUPQ design achieves performance very close to the asymptotically optimal ECUPQ, while at low rates it significantly outperforms all previous UPQ schemes. The experimental results of the proposed FRUPQ algorithm show that our approach outperforms the previous tractable designs when the total number of quantizer levels ranges between 25 and 256.

This chapter is organized as follows. The next section introduces the necessary definitions and notations. Section 2.2 formulates the problem of optimal ECUPQ design and presents the proposed solution algorithm. The problem of optimal FRUPQ design and its solution are presented in Section 2.3. The experimental results and their discussion follow in Section 2.4, while Section 2.5 concludes this chapter.

2.1 Notations

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 (Pearlman and Gray, 1978). Also, the probability density function of prediction error signal in a differential pulse code modulation coder for moving pictures can be modeled as Gaussian (Vogel, 1995).

Let M denote the number of magnitude levels of the UPQ and let $\mathbf{r} \triangleq (r_0, r_1, \cdots, r_M)$ denote the vector of thresholds of the magnitude quantizer, where

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

In this thesis, we use interchangeably the terms vector of thresholds and quantizer (or encoder) partition.

For $1 \leq m \leq M$[1], let C_m denote the m-th cell (or bin) of the magnitude quantizer, i.e., $C_m = \{r | r_{m-1} \leq r < r_m\}$. 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, $1 \leq m \leq M$. Each phase quantizer is uniform, consequently, each quantization bin of the UPQ can be represented as

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

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

[1]Note that the number of magnitude levels has to be finite in the fixed-rate case, while in the entropy-constrained case it may be infinite. However, following the prior work on the ECUPQ design, we only consider a finite number of magnitude levels.

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 (Senge, 1977; Gallagher, 1978; Wilson, 1980)

$$\begin{aligned} D &= \frac{1}{2}\sum_{m=1}^{M}\sum_{s=1}^{P_m}\int_{r_{m-1}}^{r_m}\int_{(s-1)\frac{2\pi}{P_m}}^{s\frac{2\pi}{P_m}} \|re^{j\theta} - A_m e^{j\theta_{m,s}}\|^2 p(r,\theta)d\theta dr \\ &= \frac{1}{2}\sum_{m=1}^{M}\sum_{s=1}^{P_m}\int_{r_{m-1}}^{r_m}\int_{(s-1)\frac{2\pi}{P_m}}^{s\frac{2\pi}{P_m}} (r^2 + A_m^2 - 2rA_m\cos(\theta-\theta_{m,s}))\frac{g(r)}{2\pi}d\theta dr. \end{aligned} \tag{2.1}$$

The best reconstruction values, which minimize the distortion, were determined in prior work (Senge, 1977; Gallagher, 1978; Wilson, 1980) by solving $\partial D/\partial\theta_{m,s} = 0$ and $\partial D/\partial A_m = 0$, leading to

$$\theta_{m,s} = (2s-1)\pi/P_m, \tag{2.2}$$

$$A_m = sinc\left(\tfrac{1}{P_m}\right)\frac{\int_{r_{m-1}}^{r_m} rg(r)dr}{\int_{r_{m-1}}^{r_m} g(r)dr}, \tag{2.3}$$

where $sinc(\frac{1}{P_m}) = \frac{\sin(\pi/P_m)}{\pi/P_m}$. By exploiting (2.2) and (2.3), the expected distortion can be simplified as

$$\begin{aligned} D &= \frac{1}{2}\left(\sum_{m=1}^{M}\int_{r_{m-1}}^{r_m} r^2 g(r)dr - \sum_{m=1}^{M} A_m^2\int_{r_{m-1}}^{r_m} g(r)dr\right) \\ &= \frac{1}{2}\left(\int_0^{\infty} r^2 g(r)dr - \sum_{m=1}^{M} A_m^2\int_{r_{m-1}}^{r_m} g(r)dr\right). \end{aligned} \tag{2.4}$$

Notice that, since the reconstruction values of the UPQ are given by (2.2) and (2.3), it follows that the tuples $\mathbf{r}$ and $\mathbf{P}$ completely specify the UPQ.

We point out that the proposed algorithms of UPQs are under the constraint that the magnitude quantizers' thresholds 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.

In the following section we formulate the problem of optimal ECUPQ design and propose a solution algorithm. The counterpart for the FR case is addressed in Section 2.3.

2.2 Optimal ECUPQ Design Algorithm

2.2.1 Problem Formulation

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

$$\begin{aligned} H(I_a, I_\theta) &= H(I_a) + H(I_\theta|I_a) \\ &= \sum_{m=1}^{M} q(C_m)(-\log_2 q(C_m) + \log_2 P_m), \end{aligned} \tag{2.5}$$

where for $C \subseteq \mathbb{R}$, $q(C) = \int_C g(r)dr$. Then the entropy of the UPQ (in bits/sample) is defined $H(I_a, I_\theta)/2$.

Following prior work on entropy-constrained quantization (Chou *et al.*, 1989; Muresan and Effros, 2008; Vafin and Kleijn, 2005) we formulate the problem of

ECUPQ design as follows[2]

$$\min_{M,\mathbf{r},\mathbf{P}} \quad \mathcal{L}(\mathbf{r},\mathbf{P},\lambda), \tag{2.6}$$

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 (Everett III, 1963; Luenberger, 1997) that the set of solutions to problem (2.6), 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 (2.6) minimizes the distortion for the corresponding entropy, thus it is an ECUPQ[3].

Considering that the magnitude thresholds will take values in finite set $\mathcal{A}$, the problem that we will solve in this section is the following

$$\min_{M,\mathbf{r},\mathbf{P}} \quad \mathcal{L}(\mathbf{r},\mathbf{P},\lambda), \tag{2.7}$$
$$\text{subject to } r_i \in \mathcal{A}, 1 \leq i \leq M-1.$$

2.2.2 Graph Model

In this subsection we show how the minimization problem (2.7) 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 (2.4) is constant, therefore we can remove it from the cost function. Thus, minimizing $\mathcal{L}(\mathbf{r},\mathbf{P},\lambda)$ is

[2]Note that the minimum may be achieved by a configuration with infinite M.

[3]As the Lagrangian formulation is heavily utilized throughout this thesis, the relation between the formulation of the optimal quantizer design problem as a constrained optimization problem and the corresponding Lagrangian relaxation is explained in more detail in appendix A.

equivalent to minimizing $\mathcal{F}(\mathbf{r}, \mathbf{P})$, where

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

Further, substituting (2.3) and (2.5) into the above equation leads to

$$\mathcal{F}(\mathbf{r}, \mathbf{P}) = \frac{1}{2}\sum_{m=1}^{M} \int_{r_{m-1}}^{r_m} g(r)dr \left(-sinc^2\left(\frac{1}{P_m}\right) x^2(C_m) + \lambda \log_2 \frac{P_m}{\int_{r_{m-1}}^{r_m} g(r)dr}\right), \quad (2.8)$$

where for $C \subseteq \mathbb{R}$, $x(C) = \frac{\int_C r g(r)dr}{\int_C g(r)dr}$.

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, $1 \leq m \leq M$, is

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

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

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

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

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

$$P_{[\alpha,\beta)}^* \triangleq \min\arg\min_P \left(-sinc^2\left(\frac{1}{P}\right)(x[\alpha, \beta))^2 + \lambda \log_2 P\right), \quad (2.9)$$

where the minimization is over all positive integers P[4]. Note that, if there are more values P minimizing the cost in (2.9), we select the smallest one as $P_{[\alpha,\beta)}^*$.

Further, by replacing P_m in (2.8) by $P_{[r_{m-1},r_m)}^*$, we obtain a new cost function

[4]The fact that the minimum in (2.9) is achieved follows according to Lemma 2.1 in the following section.

which only depends on $\mathbf{r}$

$$\bar{\mathcal{F}}(\mathbf{r}) \triangleq \frac{1}{2}\sum_{m=1}^{M} q[r_{m-1}, r_m)\left(\lambda \log_2 \frac{P^*_{[r_{m-1},r_m)}}{q[r_{m-1}, r_m)} - sinc^2\left(\frac{1}{P^*_{[r_{m-1},r_m)}}\right)(x[r_{m-1}, r_m))^2\right). \tag{2.10}$$

According to the above discussion, problem (2.7) is equivalent to the following

$$\begin{aligned} &\min_{M,\mathbf{r}} \quad \bar{\mathcal{F}}(\mathbf{r}) \\ &\text{subject to } r_i \in \mathcal{A}, 1 \le i \le M-1. \end{aligned} \tag{2.11}$$

The next step is based on the observation that the cost $\bar{\mathcal{F}}(\mathbf{r})$ 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,

$$w(u, v) \triangleq \frac{1}{2} q[a_u, a_v)\left(-sinc^2\left(\frac{1}{P^*_{[a_u,a_v)}}\right)(x[a_u, a_v))^2 + \lambda \log_2 \frac{P^*_{[a_u,a_v)}}{q[a_u, a_v)}\right), \tag{2.12}$$

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 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 $t_{m-1} < t_m$, $1 \le m \le s$, for some $s \ge 1$. Note that s equals the number

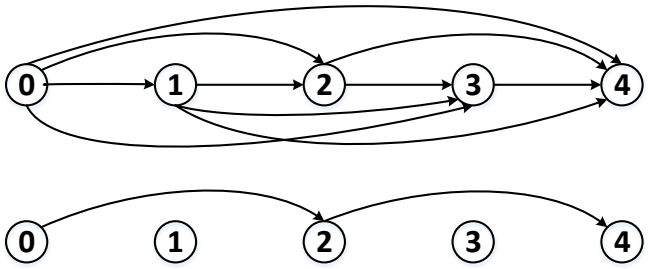


Figure 2.1: Illustration of the graph G (top) for $K = 3$ and a path in the graph (bottom). Nodes are depicted with circles and edges with arcs. The path shown on the bottom corresponds to the magnitude quantizer with bins $[0, a_2)$ and $[a_2, \infty)$.

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 $\bar{\mathcal{F}}(\mathbf{r})$. Additionally, the above correspondence is one-to-one. Therefore, we conclude that problem (2.11) 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 2.1 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 (2.9). Therefore, we have to solve problem (2.9) for all the edges. In the next subsection we present an efficient way to accomplish this goal.

2.2.3 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 (2.9) for all the edges and stores the results. First we derive an important property of the optimal number of phase regions $P^*_{[a_u,a_v)}$, based on which an efficient search strategy can be developed.

Let us denote $\mathcal{P} \triangleq \mathbb{Z}_+$. 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

$$\min_{P\in\mathcal{P}}(f(P) + \mu g(P)), \tag{2.13}$$

where $\mu > 0$. In view of (2.9), it can be easily verified that $P^*_{[a_u,a_v)}$ is a solution to problem (2.13) for $\mu = \frac{\lambda}{(x[a_u,a_v))^2 \ln 2}$.

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

$$P^*_{[a_u,a_v)} \leq P_{max}, \text{ for all } (u,v) \in E. \tag{2.14}$$

We will explain later how to find such a value. The straightforward approach to solve (2.13) is by computing the cost for each value of P, $1 \leq P \leq P_{max}$, 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

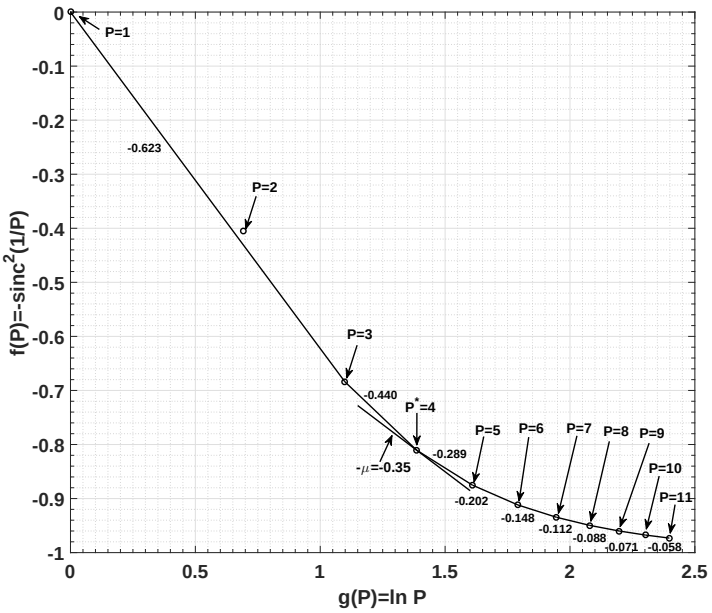


Figure 2.2: Illustration of the set $\mathcal{U}$ of points of coordinates $(g(P), f(P))$, and of the set $\hat{\mathcal{U}}$, the lower boundary of the convex hull of $\mathcal{U}$. The number near each convex hull edge represents its slope. When $\mu = 0.35$ the solution to problem (2.13) is $P^* = 4$ 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.

(2.13).

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 (Everett III, 1963; Luenberger, 1997) that some value P^* minimizes the cost in (2.13) 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 2.2 illustrates the sets $\mathcal{U}$ and $\hat{\mathcal{U}}$. It also shows that $P^* = 4$ is the solution to problem (2.13) 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. Note that $left\_slope(1) = -\infty$. Then the following relation holds

$$left\_slope(P) \leq right\_slope(P), \text{ for any } P \in \hat{\mathcal{P}}. \tag{2.15}$$

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 (2.15) is verified in Figure 2.2.

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

$$left\_slope(P) \leq -\mu \leq right\_slope(P). \tag{2.16}$$

In light of the above discussion[5], we obtain the following characterization of $P^*_{[a_u, a_v)}$, stated as a lemma. Its proof is deferred to Appendix B.

[5]A rigorous proof of the points highlighted above is provided in Appendix B.

Lemma 2.1. For each $(u, v) \in E$, a value $P \in \hat{\mathcal{P}}$ satisfying

$$left\_slope(P) \leq -\frac{\lambda}{(x[a_u, a_v))^2 \ln 2} \leq right\_slope(P) \tag{2.17}$$

always exists, and the smallest such P equals $P^*_{[a_u,a_v)}$. The above lemma together with (2.15) implies that $P^*_{[a_u,a_v)}$ 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 appendix B.

Proposition 2.1. $\hat{\mathcal{P}} = \mathcal{P} \setminus \{2\}$.

Note that Figure 2.2 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 2.2. For any integers u, u', v, v' such that $0 \leq u < v \leq K+1$, $0 \leq u' < v' \leq K+1$, and such that $u \leq u'$ and $v \leq v'$, the following inequality holds:

$$P^*_{[a_u,a_v)} \leq P^*_{[a_{u'},a_{v'})}. \tag{2.18}$$

Proof:

Notice that $x[\alpha, \beta)$ is the centroid of the interval $[\alpha, \beta)$. It is known that $x[\alpha, \beta) \geq$

α and that $x[\alpha, \beta)$ is a non-decreasing function of both α and β[6]. Then, under the conditions specified in the hypothesis, it follows that

$$0 \leq x[a_u, a_v) \leq x[a_{u'}, a_{v'}),$$

which further leads to

$$-\frac{\lambda}{(x[a_u, a_v))^2 \ln 2} \leq -\frac{\lambda}{(x[a_{u'}, a_{v'}))^2 \ln 2}, \tag{2.19}$$

since λ and $\ln 2$ are positive. The above inequality together with Lemma 2.1 and relations (2.15) implies inequality (2.18), thus proving the claim.

Remark. Proposition 2.2 implies that

$$P^*_{[a_u, a_v)} \leq P^*_{[a_K, a_{K+1})} \text{ for all } 0 \leq u < v \leq K+1.$$

Then the value of $P^*_{[a_K, a_{K+1})}$ can be set as P_{max}. In view of Lemma 2.1 and Proposition 2.1, the value $P^*_{[a_K, a_{K+1})}$ can be determined by inspecting all positive integers $P, P \neq 2$, in increasing order until relation (2.17) is satisfied.

Proposition 2.2 implies that the search range for $P^*_{[a_u, a_v)}$ can be reduced from $\hat{\mathcal{P}}$ to the smaller set $[P^*_{[a_u, a_{v-1})}, P^*_{[a_{u+1}, a_v)}] \cap \hat{\mathcal{P}}$, if $P^*_{[a_u, a_{v-1})}$ and $P^*_{[a_{u+1}, a_v)}$ are evaluated first. Therefore, in order to exploit this observation we need to choose carefully the order of computation of the values $P^*_{[a_u, a_v)}$. To facilitate a visual representation of this ordering imagine that $P^*_{[a_u, a_v)}$ is the element on row u and column v of an upper triangular matrix 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 $P^*_{[a_u, a_{u+1})}$,

[6] A proof of this result can be found in (Trushkin, 1982).

Algorithm 1: Efficient procedure to precompute all values $P^*_{[a_u,a_v)}$.

```
for v = 1 to K + 1 do
    P*_[a_{v-1},a_v) := min arg min_{P∈P̂} E(P, v − 1, v)
    for u = v − 2 down to 0 do
        P*_[a_u,a_v) := min arg min_{P*_[a_u,a_{v-1})≤P≤P*_[a_{u+1},a_v)} E(P, u, v)
```

$0 \leq u \leq K$. We will compute the elements of this upper triangular matrix starting in the top left corner, i.e., with $P^*_{[a_0,a_1)}$, 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

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

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 $P^*_{[a_u,a_{u+j+1})}$, $0 \leq u \leq K-j$. The entry $P^*_{[a_u,a_{u+j+1})}$ is evaluated in $O(P^*_{[a_{u+1},a_{u+j+1})} - P^*_{[a_u,a_{u+j})} + 1)$ time. Therefore, the total time for the j-th superdiagonal is

$$\begin{aligned} &O\left(\sum_{u=0}^{K-j}\left(P^*_{[a_{u+1},a_{u+j+1})} - P^*_{[a_u,a_{u+j})} + 1\right)\right) \\ &= O\left(P^*_{[a_{K-j+1},a_{K+1})} - P^*_{[a_0,a_j)} + K - j + 1\right) \\ &= O\left(P_{max} + K\right). \end{aligned}$$

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,

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

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

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

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

$$q[a_u, a_v) = \phi_0(v) - \phi_0(u),$$
$$x[a_u, a_v) = \frac{\phi_1(v) - \phi_1(u)}{q[a_u, a_v)}.$$

Recall that if all values $P^*_{[a_u, a_v)}$ 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 $P^*_{[a_u, a_v)}$ 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 $P^*_{[a_u,a_v)}$ 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 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 (2.7) including the above procedure for determining the values $P^*_{[a_u,a_v)}$. We point out that $\hat{W}(v)$ denotes the weight of the MWP 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 $P_{max} = P^*_{[a_K,a_{K+1})}$, and the cumulative probabilities and first moments are computed and stored.

In conclusion, solving problem (2.7) 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 (2.7) is $O(K^2)$.

2.3 Optimal FRUPQ Design Algorithm

2.3.1 Problem Formulation

For each positive integer k and extended real number $\beta \in \mathcal{A} \cap (\mathbb{R} \cup \{\infty\})$, denote by $\mathfrak{T}_k(\beta)$ the set of all vectors of thresholds $\mathbf{r} = (r_0, r_1, \cdots, r_k)$ such that $0 = r_0 < r_1 < r_2 < \cdots < r_k = \beta$ and $r_m \in \mathcal{A}$ for all $1 \leq m \leq k-1$.

The problem of FRUPQ design can be formulated as the following level-constrained

Algorithm 2: Solution algorithm for problem (2.7).

Preprocessing Stage
begin
- $\hat{W}(0) = 0$
- **for** $v = 1$ **to** $K + 1$ **do**
  - Allocate memory of size v to store $P^*_{[\cdot,a_v)}$
  - $P^*_{[a_{v-1},a_v)} := \min \arg\min_{P\in\hat{\mathcal{P}}} E(P, v-1, v)$
  - $\hat{W}(v) := \hat{W}(v-1) + w(v-1, v)$
  - $\varepsilon(v) := v - 1$
  - **for** $u = v - 2$ **down to** 0 **do**
    - $P^*_{[a_u,a_v)} := \min \arg\min_{P^*_{[a_u,a_{v-1})}\le P\le P^*_{[a_{u+1},a_v)}} E(P, u, v)$
    - **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 $P^*_{[\cdot,a_{v-1})}$
- *// 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** *(*$i \le j$*)* **do**
  - $t_i := s(j - i)$
  - $i := i + 1$

minimization problem

$$\begin{min}_{M,\mathbf{r},\mathbf{P}} \quad D \\ \text{subject to } \sum_{m=1}^{M} P_m = N,\ P_m \in \mathbb{Z}_+,\ \mathbf{r} \in \mathcal{T}_M(\infty),\end{min} \tag{2.20}$$

where $\mathbb{Z}_+$ is the set of positive integers and N is the target value for the number of levels of the UPQ. In this section we propose a globally optimal solution to the above problem.

2.3.2 Dynamic Programming Solution

In this subsection we present a solution to problem (2.20) based on dynamic programming. First we will introduce a few more notations. For $\alpha \leq \beta$ and positive integer P denote

$$\omega_P(\alpha, \beta) \triangleq \frac{1}{2} f(P) \left(x[\alpha, \beta)\right)^2 q[\alpha, \beta). \tag{2.21}$$

Notice that the first term of the distortion formulation in (2.4) is constant, therefore it can be removed from the cost function of (2.20). After doing so the objective function of (2.20) becomes

$$\mathcal{O}(\mathbf{r}, \mathbf{P}) \triangleq \sum_{m=1}^{M} \omega_{P_m}(r_{m-1}, r_m).$$

For each pair of positive integers (k, n) with $1 \leq k \leq K+1$ and $1 \leq n \leq N$, consider problem $\mathcal{P}(k, n)$ defined as

$$\begin{min}_{M,\mathbf{r},\mathbf{P}} \quad \mathcal{O}(\mathbf{r}, \mathbf{P}) \\ \text{subject to } \sum_{m=1}^{M} P_m = n,\ P_m \in \mathbb{Z}_+,\ \mathbf{r} \in \mathcal{T}_M(a_k).\end{min} \tag{2.22}$$

Additionally, denote by $\hat{\mathcal{O}}(k,n)$ the optimal value of the objective function in (2.22), for $1 \leq k \leq K+1$ and $1 \leq n \leq N$.

Intuitively, problem (2.22) can be interpreted as finding the optimal FRUPQ with n levels, corresponding to the portion of the magnitude space ranging from 0 to a_k. It can be easily seen that problem (2.20) is equivalent to $\mathcal{P}(K+1,N)$. The dynamic programming solution consists of solving all sub-problems $\mathcal{P}(k,n)$, for $1 \leq k \leq K+1$ and $1 \leq n \leq N$, using the following recurrence relation

$$\hat{\mathcal{O}}(k,n) = \min_{0 \leq t < n} \min_{0 \leq j < k} \left(\hat{\mathcal{O}}(j,t) + \omega_{n-t}(a_j, a_k) \right), \tag{2.23}$$

where $\hat{\mathcal{O}}(0,0) = 0$ and $\hat{\mathcal{O}}(0,t) = \hat{\mathcal{O}}(j,0) = \infty$, for $t > 0$ and $j \geq 1$. The dynamic programming process evaluates (2.23) in increasing order of k and n. For each pair (k,n) the minimizations in (2.23) take $O(KN)$ operations if each quantity $\omega_{n-t}(a_j, a_k)$ can be evaluated in constant time. Since there are $O(KN)$ pairs (k,n) in total, the time complexity of the solution algorithm becomes $O(K^2N^2)$. It can be seen from (2.21) that for computing the values $\omega_{n-t}(a_j, a_k)$ the quantities $x[a_j, a_k)$ and $q[a_j, a_k)$ are needed. In order to enable the computation of each $x[a_j, a_k)$ and $q[a_j, a_k)$ in constant time, the cumulative probabilities and first moments are precomputed and stored in a preprocessing step as in Section 2.2.3, which only requires $O(K)$ operations.

In the next subsection we will show that the algorithm can be sped up by exploiting a certain monotonicity property of the objective function.

2.3.3 Complexity Reduction

For each pair of integers (n, t) with $1 \leq t < n \leq N$, consider the upper triangular matrix $G_{n,t}$ with elements $G_{n,t}(j, k)$, $1 \leq j < k \leq K + 1$,

$$G_{n,t}(j, k) \triangleq \hat{\mathcal{O}}(j, t) + \omega_{n-t}(a_j, a_k). \tag{2.24}$$

Clearly, the minimization over j in (2.23) is equivalent to finding the smallest element on column k of matrix $G_{n,t}$, i.e., finding

$$\hat{G}_{n,t}(k) \triangleq \min_{1 \leq j \leq k+1} G_{n,t}(j, k). \tag{2.25}$$

Then relation (2.23) is equivalent to

$$\hat{\mathcal{O}}(k, n) = \min \left(\omega_n(0, a_k), \min_{1 \leq t < n} \hat{G}_{n,t}(k) \right). \tag{2.26}$$

Determining all column minima takes $O(K^2)$ time in a general $O(K)$-by-$O(K)$ matrix. However, when the matrix is *totally monotone* this task can be accomplished in $O(K)$ time using the algorithm nicknamed SMAWK (Aggarwal *et al.*, 1987). According to (Aggarwal *et al.*, 1987) matrix $G_{n,t}$ is said to be totally monotone (with respect to the column minima problem[7]) if for all $j < j'$ and $k < k'$ the following implication holds

$$G_{n,t}(j', k) < G_{n,t}(j, k) \Rightarrow G_{n,t}(j', k') < G_{n,t}(j, k').$$

[7]The total monotonicity is defined in (Aggarwal *et al.*, 1987) for the problem of row maxima, which can be converted to the column minima problem by transposing the matrix and multiplying all entries by -1. Here we adapt the definition of total monotonicity to the column minima problem.

A sufficient condition for the total monotonicity to hold is the following, known as the *Monge* condition (Burkard *et al.*, 1996)

$$G_{n,t}(j,k) + G_{n,t}(j',k') \leq G_{n,t}(j,k') + G_{n,t}(j',k) \tag{2.27}$$

for all $1 \leq j < j' < k < k' \leq K+1$.

Proposition 2.3. Matrix $G_{n,t}$ satisfies the Monge condition.

Proof:

By replacing (2.24) in (2.27) and performing the cancellation of the like terms, (2.27) becomes equivalent to

$$\omega_{n-t}(a_j, a_k) + \omega_{n-t}(a_{j'}, a_{k'}) \leq \omega_{n-t}(a_j, a_{k'}) + \omega_{n-t}(a_{j'}, a_k). \tag{2.28}$$

Define now, for $1 \leq j < k \leq K+1$,

$$d(j,k) \triangleq \int_{a_j}^{a_k} r^2 g(r) dr - \left(x[a_j, a_k)\right)^2 q[a_j, a_k).$$

It was shown in (Wu, 1991) that $d(j,k)$ satisfies the Monge condition, i.e., the following holds

$$d(j,k) + d(j',k') \leq d(j,k') + d(j',k), \tag{2.29}$$

for all $1 \leq j < j' < k < k' \leq K+1$. Note from (2.21) that

$$d(j,k) = \int_{a_j}^{a_k} r^2 g(r) dr + \frac{2}{sinc^2(\frac{1}{n-t})} \omega_{n-t}(a_j, a_k).$$

By applying the above in (2.29) and performing some algebraic manipulations, relation (2.28) follows.

The fast solution algorithm proceeds as follows. It iterates over n in increasing

order from 1 to N. For each n, problem $\mathcal{P}(k,n)$ is solved for all k, as follows. We increase t from 1 to $n-1$ and for each t all column minima in matrix $G_{n,t}$ are determined using SMAWK. This requires $O(K)$ time for each matrix. Over all values of t, this amounts to $O(KN)$ operations. After that the minimization over t in (2.23) is performed, for each k, requiring a total of $O(KN)$ operations. Performing the above for all n leads to $O(KN^2)$ time complexity for the solution algorithm.

Note that in order to apply SMAWK, the matrix $G_{n,t}$ has to be extended to a full matrix. This can be done by setting to ∞ all elements below the main diagonal. This extension does not change the column minima, and the full matrix still satisfies the total monotonicity (Burkard *et al.*, 1996).

The following pseudocode (Algorithm 3) describes the algorithm to solve problem (2.20). We use the notation $\hat{j}_{n,t}(k)$ for the value of j achieving optimality in (2.25), and $\hat{t}(n,k)$ for the optimal t in (2.23).

2.4 Experimental Results

This section assesses the practical performance of the proposed UPQ design algorithms. 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

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

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

Algorithm 3: Solution algorithm to problem (2.20).

Preprocessing Stage
begin
  for $k = 1$ **to** $K + 1$ **do**
    $\hat{\mathcal{O}}(k, 1) = \omega_1(0, a_k)$ /* $n = 1$ */
    $\hat{j}_{1,0}(k) = 0$
    $\hat{t}(1, k) = 0$
  for $n = 1$ **to** N **do**
    $\hat{\mathcal{O}}(1, n) = \omega_n(0, a_1)$ /* $k = 1$ */
    $\hat{j}_{n,0}(1) = 0$
    $\hat{t}(n, 1) = 0$
  for $n = 2$ **to** N **do**
    for $t = 1$ **to** $n - 1$ **do**
      Evaluate $\hat{G}_{n,t}(k)$ for all k using SMAWK
      Record $\hat{j}_{n,t}(k)$ for all k
    for $k = 2$ **to** $K + 1$ **do**
      Compute $\hat{\mathcal{O}}(k, n)$ using (8)
      Record $\hat{t}(n, k)$
  Restore the vectors r and P

Rate	$10 \log_{10} D$	$10 \log_{10} D^{(\text{Wilson, 1980})}$	$10 \log_{10} \frac{D^{(\text{Wilson, 1980})}}{D}$	$10 \log_{10} \frac{D}{D_G(R)}$
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

Table 2.1: Performance comparison of the proposed ECUPQ with the entropy-coded UPQ of (Wilson, 1980) and $D_G(R)$, for rates $R < 2.5$ bits/sample.

We first consider the case of ECUPQ, where we compare the proposed algorithm with the designs of (Wilson, 1980), (Vafin and Kleijn, 2005) and with entropy-constrained rectangular quantizer (ECRQ). 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 (2.4). 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 (Wilson, 1980) is performed for rates in the range from 0.5 to 2.5 bits/sample, based on the results reported in (Wilson, 1980). The comparison with the asymptotically optimal ECUPQ of (Vafin and Kleijn, 2005) is performed for rates higher than 2.05.

Table 2.1 illustrates the performance comparison with (Wilson, 1980). Recall that the UPQ of (Wilson, 1980) is fixed-rate, 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 (Wilson, 1980) are taken from (Wilson, 1980). The second last column in the table shows the gain in performance of the proposed approach versus the method of (Wilson, 1980). It can be seen that our algorithm always outperforms the design of (Wilson, 1980) 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 2.1 lists the gap between the ECUPQ distortion and the

Rate	$(N, M, P_1, \cdots, P_M)$	$(r_1, \cdots, r_{M-1})$	$(N, M, P_1, \cdots, P_M)^{\text{(Wilson, 1980)}}$	$(r_1, \cdots, r_{M-1})^{\text{(Wilson, 1980)}}$	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 2.2: Configuration of the proposed ECUPQ, of the entropy-coded UPQ of (Wilson, 1980) and of the optimal ECRQ, for rates $R < 2.5$ bits/sample.

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 $(r_1, \cdots, r_{M-1})$ and the configurations $(N, M, P_1, \cdots, P_M)$ for the proposed ECUPQ and for the UPQ of (Wilson, 1980) are presented in Table 2.2. We observe that for the same output entropy the fixed-rate UPQ of (Wilson, 1980) 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 2.3a), while for the UPQ of (Wilson, 1980), 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 2.2. 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 (Vafin and Kleijn, 2005) based on the high resolution assumption. We will use the acronym ASY to refer to the asymptotical ECUPQ performance derived

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 2.3: 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.

in (Vafin and Kleijn, 2005). Note that the asymptotical distortion (per sample) of ASY obtained in (Vafin and Kleijn, 2005) is

$$D_{ASY} = \frac{2^{-(2R-\log_2(2\pi e))}}{12}, \tag{2.30}$$

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

Table 2.3 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 2.3 also shows the gap between the performance of the proposed ECUPQ and the asymptotical performance (per sample) of the two-dimensional entropy-constrained

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

Table 2.4: Performance comparison of the proposed ECUPQ with PASY.

vector quantizer (ECVQ), computed based on (Gersho, 1979)

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

Moreover, the difference in performance versus the distortion-rate function is also presented in Table 2.3. 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 (Vafin and Kleijn, 2005). 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 corresponding quantization step

sizes, are derived in (Vafin and Kleijn, 2005) as

$$g_A(a) = \sqrt{\frac{1}{6\lambda \log_2(e)}}, \tag{2.31}$$

$$g_\Theta(\theta, a) = \sqrt{\frac{a^2}{6\lambda \log_2(e)}}, \tag{2.32}$$

where $\lambda > 0$ is the Lagrangian multiplier and a denotes the reconstructed magnitude value. Notice that the magnitude quantizer corresponding to (2.31) is uniform with step size $1/g_A(a)$, while the phase quantizer corresponding to (2.32) has step size $1/g_\Theta(\theta, a)$.

To implement the UPQ based on equations (2.31) and (2.32) we proceed as follows. The vector of thresholds $\mathbf{r}$ 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 (2.2). We evaluate the distortion and entropy of PASY using (2.1), respectively, $H(I_a, I_\theta)/2$, where $H(I_a, I_\theta)$ is given in (2.5).

Table 2.4 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.

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

Table 2.5: Performance comparison of the proposed ECUPQ against ECRQ.

Since the authors of (Vafin and Kleijn, 2005) 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 (Muresan and Effros, 2008). We point out that the algorithm of (Muresan and Effros, 2008) 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 2.2 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

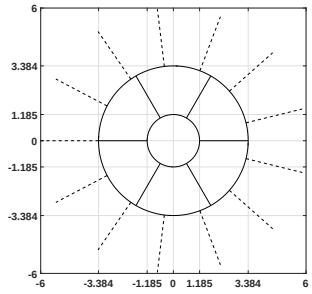
between ECUPQ and ECRQ is illustrated in Table 2.5 only for the range of rates from 0.5 to 2.422, since for higher rates the absolute value of the performance difference is less than 0.01 dB. The results in Table 2.5 show that the proposed ECUPQ outperforms ECRQ in the low-rate region with improvements 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 2.3a for the ECUPQ at rate 1.157 and in Figure 2.3b for the ECRQ at the same rate. Two possible reasons for the superiority of ECUPQ at low rates are:

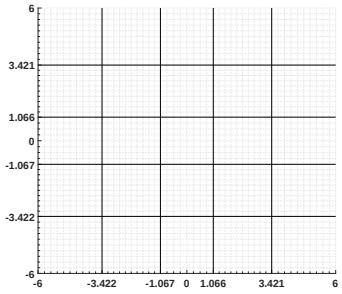
1) ECUPQ has higher flexibility in the choice of the number N of quantization bins, as it can be seen in Table 2.2. Namely, for ECUPQ N could be any positive integer, while for ECRQ N^{ECRQ} can only be a perfect square. We see that the ECUPQ with rate 1.157 has $N = 20$, while the ECRQ cannot select such a value for N^{ECRQ}. Instead it has $N^{ECRQ} = 25$. The same conclusion holds for all the rates illustrated in Table 2.2.

2) We observe in Figure 2.3a 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 2.2, we see that the proposed ECUPQ always has a disc-shaped cell around the origin.

It is also interesting to compare the proposed ECUPQ design with the practical two-dimensional vector quantizer (2DVQ). For this purpose, we have used the "vqdtool" in MATLAB R2018a to obtain the practical 2DVQ. The training set contains



(a) ECUPQ.



(b) ECRQ.

Figure 2.3: The partitions of proposed ECUPQ (a) and ECRQ (b) at rate $R = 1.157$ bits/sample.

10^7 two-dimensional Gaussian source vectors, where the two scalar components are independent and each has zero-mean and unit-variance. The initial codebook is generated automatically. The stopping criteria is that the relative decrease of the squared error has to be smaller than 10^{-7}. Further, when a training vector has the same distortion for two different codewords, the lower indexed codeword will be selected. The design is for the fixed-rate case, but entropy coding is applied to the output of the 2DVQ, where the term entropy-coded 2DVQ is used.

Table 2.6 illustrates the performance comparison between the proposed ECUPQ and the entropy-coded 2DVQ, where the distortion is denoted by D^E_{2DVQ}. The number of cells (N^{2DVQ}) of the 2DVQ is also given in the table. It can be noticed that the proposed ECUPQ always outperforms the entropy-coded 2DVQ, reaching a peak improvement of 0.639 dB at a rate of 2.899 bits/sample. The large improvement can be explained by the fact that the 2DVQ is optimized under the constraint that the number of cells is fixed, instead of imposing a constraint on the entropy. Another reason could be attributed to the local optimality of the 2DVQ design algorithm.

Rate	$10\log_{10} D$	$10\log_{10} D^{E}_{2DVQ}$	N^{2DVQ}	$10\log_{10} \frac{D^{E}_{2DVQ}}{D}$
0.500	-2.127	-1.665	2	0.462
1.000	-4.692	-4.398	4	0.294
1.460	-7.358	-6.918	8	0.440
1.946	-10.217	-9.642	16	0.575
2.418	-13.036	-12.430	32	0.606
2.899	-15.902	-15.263	64	0.639
3.385	-18.778	-18.172	128	0.606
3.878	-21.375	-21.115	256	0.260

Table 2.6: Performance comparison of the proposed ECUPQ against entropy-coded 2DVQ.

To summarize, we conclude that the proposed ECUPQ design algorithm outperforms the algorithm of (Wilson, 1980) 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.

Next we assess the performance of the proposed FRUPQ design algorithm in comparison with the designs of (Perić and Nikolić, 2013), (Wilson, 1980) and (Petković *et al.*, 2011). In the FRUPQ case, we take $K = 3000$ and $a_i = 0.002i$, for $1 \leq i \leq K$. We applied the dynamic programming algorithm to construct the optimal FRUPQ with $N = 256$. The FRUPQs for all $N < 256$ were also generated during the dynamic programming process.

Wilson (Wilson, 1980) constructed the optimal FRUPQs for all N between 1 and 16, and for $25, 32$ and 36, and reported the optimal configuration $M, \mathbf{P}, \mathbf{r}$. Our

N	$(M, P_1, \cdots, P_M)$	$(r_1, \cdots, r_{M-1})$	$10\log_{10} D$	$(M, P_1, \cdots, P_M)^{(\text{Wilson, 1980})}$	$(r_1, \cdots, r_{M-1})^{(\text{Wilson, 1980})}$	$10\log_{10} D^{(\text{Wilson, 1980})}$	$10\log_{10} \frac{D^{(\text{Wilson, 1980})}}{D}$
25	$(3, 5, 9, 11)$	$(0.856, 1.682)$	-11.3188	$(3, 4, 10, 11)$	$(0.798, 1.674)$	-11.3181	0.0007
36	$(4, 3, 8, 12, 13)$	$(0.524, 1.130, 1.898)$	-12.7805	$(4, 1, 8, 13, 14)$	$(0.369, 1.051, 1.848)$	-12.7772	0.0033

Table 2.7: Performance comparison with the FRUPQ of (Wilson, 1980) and the corresponding optimal configuration, for $N = 25$ and 36.

N	$(M, P_1, \cdots, P_M)$	$(r_1, \cdots, r_{M-1})$	$10\log_{10} D$	$(M, P_1, \cdots, P_M)^{(\text{Petković } et\ al., 2011)}$	$(r_1, \cdots, r_{M-1})^{(\text{Petković } et\ al., 2011)}$	$10\log_{10} D^{(\text{Petković } et\ al., 2011)}$	$10\log_{10} \frac{D^{(\text{Petković } et\ al., 2011)}}{D}$
64	$(5, 5, 10, 15, 18, 16)$	$(0.536, 0.998, 1.534, 2.234)$	-15.150	$(6, 2, 6, 11, 15, 16, 14)$	$(0.277, 0.663, 1.120, 1.655, 2.345)$	-15.082	0.068
128	$(8, 1, 7, 13, 18,$ $22, 24, 24, 19)$	$(0.180, 0.498, 0.826, 1.174,$ $1.564, 2.026, 2.650)$	-18.053	$(8, 4, 9, 14, 18,$ $22, 23, 22, 16)$	$(0.324, 0.623, 0.943, 1.290,$ $1.688, 2.161, 2.806)$	-17.991	0.062
256	$(11, 1, 8, 14, 20, 25,$ $29, 33, 35, 35, 32, 24)$	$(0.138, 0.378, 0.610, 0.848, 1.098,$ $1.364, 1.660, 1.998, 2.408, 2.972)$	-20.985	–	–	-20.907	0.078

Table 2.8: Performance comparison with the FRUPQ of (Petković *et al.*, 2011) and the corresponding optimal configuration.

approach generated the same FRUPQs as in (Wilson, 1980) for all N, except for $N = 25$ and 36. The results for the latter values and the comparison with (Wilson, 1980), are presented in Table 2.7. We see that our design exhibits an improvement in distortion of 0.0033 dB for $N = 36$, respectively 0.0007 dB for $N = 25$. It is worth pointing out that, while the performance of the FRUPQ of (Wilson, 1980) is identical or very close to our scheme for small values of N, the algorithm of (Wilson, 1980) is not tractable for larger values of N, because of the exponential growth of the space of all configurations $(M, \mathbf{P})$ satisfying $N = \sum_{m=1}^{M} P_m$. On the other hand, the time complexity of our proposed solution grows only quadratically with N, therefore it is tractable for much larger values.

Table 2.8 illustrates the comparison with the FRUPQ of (Petković *et al.*, 2011). Note that the authors of (Petković *et al.*, 2011) only report the distortions for $N = 64, 128, 256$, and the optimal FRUPQ parameters $M, \mathbf{P}, \mathbf{r}$ for $N = 64, 128$. It can be seen that our algorithm always outperforms the design of (Petković *et al.*, 2011) with gains higher than 0.06 dB, and reaching a peak improvement of 0.078 dB when $N = 256$.

Next we compare the performance of the proposed design with the FRUPQ of

N	$10\log_{10} D$	$10\log_{10} D_{ASY}$	$10\log_{10} D_{PASY}$	$10\log_{10} \frac{D_{PASY}}{D}$
16	-9.614	-9.572	-9.324	0.290
32	-12.340	-12.297	-12.206	0.134
64	-15.150	-15.125	-15.075	0.075
128	-18.053	-18.022	-17.969	0.084
256	-20.985	-20.963	-20.945	0.040

Table 2.9: Performance comparison of the proposed FRUPQ with ASY and PASY of (Perić and Nikolić, 2013), for $N \geq 16$.

(Perić and Nikolić, 2013), using the results reported in (Perić and Nikolić, 2013). We use the acronym ASY to refer to the asymptotical performance derived in (Perić and Nikolić, 2013), and the acronym PASY to refer to the practical design counterpart. Table 2.9 illustrates the performance of the proposed algorithm in comparison with ASY and PASY, for N taking as values the powers of 2 from 16 to 256. We see that the proposed algorithm is superior to both ASY and PASY for all values of N examined. Specifically, the gains over ASY are always higher than 0.01 dB, with a peak of 0.043 dB at $N = 32$. The performance improvement over PASY ranges between 0.29 dB and 0.075 dB for N between 16 and 128. Additionally, we observe that the gap between PASY and the proposed scheme tends to decrease as N increases. This is expected since PASY is globally optimal as $N \to \infty$, therefore its accuracy is expected to improve as N increases. On the other hand, since the proposed approach is globally optimal at finite rates (subject to the confined set of thresholds), it can serve as a benchmark to establish the accuracy of PASY and ASY at finite rates.

Finally, Figure 2.4 plots the distortion in dB (i.e., $10\log_{10} D$) versus rate, computed as $R = \frac{1}{2}\log_2 N$, for the proposed FRUPQ, the PASY scheme in (Perić and Nikolić, 2013) and the design of (Petković *et al.*, 2011), where the plots of (R, D) pairs at $R = 3$ and 3.5 are magnified.

Additionally, we also compare the proposed FRUPQ design with the practical

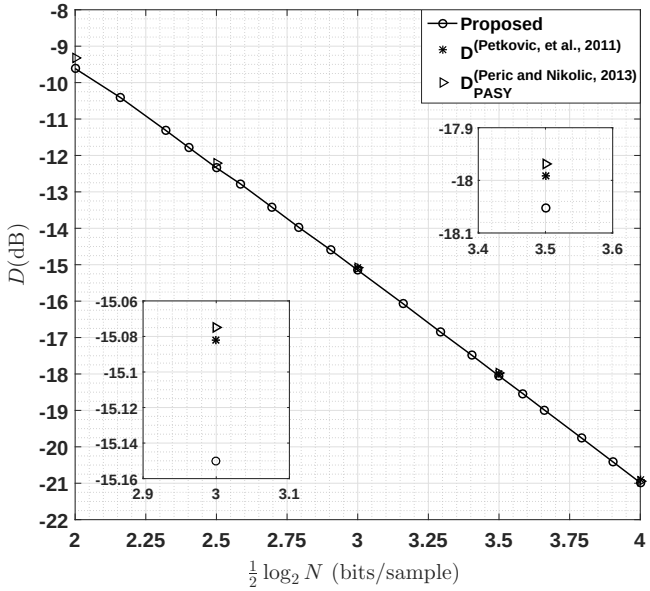


Figure 2.4: Performance comparison with PASY (Perić and Nikolić, 2013) and with (Petković *et al.*, 2011).

fixed-rate 2DVQ, whose distortion is denoted by D_{2DVQ}^{FR}. The design procedure for the 2DVQ is the same as the description above Table 2.6. Table 2.10 demonstrates the comparison between the proposed FRUPQ and the fixed-rate 2DVQ, for $N \geq 2$. It can be observed that the proposed FRUPQ performs very close to the fixed-rate 2DVQ, but with a lower coding complexity. Specifically, the gap to the fixed-rate 2DVQ is smaller than 0.03 dB for $N \leq 8$ and $N = 16$, while the largest gap is only 0.13 dB at $N = 256$.

Before ending this section we would like to briefly address the problem of choosing the set $\mathcal{A}$ of possible thresholds. Although the goals of this chapter are solving problems (2.7) and (2.20), which assume that the set $\mathcal{A}$ is given, the choice of $\mathcal{A}$ determines how well the solutions of (2.7) and (2.20) approximate the solutions to the corresponding unconstrained problems. 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

N	$10\log_{10} D$	$10\log_{10} D_{2DVQ}^{FR}$	$10\log_{10} \frac{D_{2DVQ}^{FR}}{D}$
2	-1.664	-1.665	-0.001
3	-3.346	-3.347	-0.001
4	-4.396	-4.398	-0.002
5	-5.096	-5.123	-0.027
6	-5.947	-5.964	-0.017
7	-6.518	-6.525	-0.007
8	-6.913	-6.918	-0.005
10	-7.731	-7.837	-0.106
14	-9.036	-9.128	-0.092
16	-9.614	-9.642	-0.028
32	-12.340	-12.430	-0.090
64	-15.150	-15.263	-0.113
128	-18.053	-18.172	-0.119
256	-20.985	-21.115	-0.130

Table 2.10: Performance comparison of the proposed FRUPQ against fixed-rate 2DVQ, for $N \geq 2$.

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.

2.5 Conclusion

This chapter addresses the design of UPQ for bivariate circularly symmetric sources, for both EC and FR cases. We propose design algorithms which are globally optimal

when the thresholds of the magnitude quantizer are confined to a finite set. Our solution to the ECUPQ design problem consists of solving the MWP problem in a certain WDAG, in conjunction with an efficient procedure to find the optimal number of phase regions for each possible magnitude quantizer bin. The proposed solution to the FRUPQ design problem is a dynamic programming algorithm sped up based on a monotonicity property of the objective function. 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 for the ECUPQ case. The experimental results of the proposed FRUPQ algorithm show better performance than predicted by the high-rate quantization theory and than the prior tractable designs when the number of quantizer cells ranges between 25 and 256.

Chapter 3

Design of Successively Refinable Unrestricted Polar Quantizer

This chapter addresses the design of SRUPQ with two refinement stages for bivariate circularly symmetric sources. We consider both the FR and EC cases. The proposed algorithms are globally optimal under the constraint that the magnitude quantizers' thresholds are confined to finite sets.

The optimization problem for the EC case is formulated as the minimization of a weighted sum of distortions and entropies. The proposed solution involves a series of stages including solving the MWP problem for multiple node pairs in certain WDAGs. The solution algorithm for the FR case is based on solving a series of dynamic programming problems for multiple coarse quantizer bins. The asymptotical time complexity is $O(K_1 K_2^2 P_{cmax})$ for the EC case, where K_1 and K_2 are the sizes of the sets of possible magnitude thresholds of the coarse UPQ and the refined UPQ, respectively, while P_{cmax} is the maximum number of phase levels in any phase quantizer of the coarse UPQ. The time complexity for the FR case amounts to $O(K_1 K_2 N'^2 N_1)$, where N_1 is the number of bins of the coarse UPQ, while N' denotes the ratio between

the number of bins of the fine UPQ and the coarse UPQ.

The rest of the chapter is organized as follows. The next section introduces the necessary notations. Section 3.2 formulates the problem of optimal EC-SRUPQ design and presents the proposed solution algorithm, while the formulation for the FR-SRUPQ problem and its solution are presented in Section 3.3. The experimental results and their discussion follow in Section 3.4 and finally Section 3.5 concludes this chapter.

3.1 Notations

For any integer $n \geq 2$, an ascending n-sequence (vector of n-thresholds) is an n-tuple $\mathbf{t} = (t_0, t_1, t_2, \cdots, t_{n-1})$, with $t_i \in [0, \infty)$, for $0 \leq i \leq n-2$, and $t_{n-1} \in [0, \infty]$, where $t_0 < t_1 < t_2 < \cdots < t_{n-2} < t_{n-1}$. For any $n \geq 2$, $a \in [0, \infty)$ and $b \in [0, \infty]$, with $a < b$, let $\mathcal{S}_n(a, b)$ denote the set of all ascending n-sequences such that $t_0 = a$ and $t_{n-1} = b$.

This chapter addresses the design of the SRUPQ with two refinement stages, which can be represented as an ordered pair of embedded UPQs $\mathbf{Q} = (Q_1, Q_2)$, where Q_1 is the coarse UPQ, while Q_2 is the refined UPQ.

The definition of Q_1 is the same as the UPQ with M_1 magnitude levels in Chapter 2. Recall that the vector of thresholds of the magnitude quantizer is $\mathbf{r} \triangleq (r_0, r_1, \cdots, r_{M_1})$, the i-th cell is defined by $C_i = \{r | r_{i-1} \leq r < r_i\}$ for $1 \leq i \leq M_1$, and $\mathbf{P} \triangleq (P_1, P_2, \cdots, P_{M_1})$ denotes the sequence of the number of phase regions corresponding to C_i. Then the total number of quantization bins of Q_1 is given by $N(Q_1) = \sum_{i=1}^{M_1} P_i$, $1 \leq i \leq M_1$.

The magnitude partition of the refined UPQ Q_2 is embedded in the partition $\mathbf{r}$. This means that each cell C_i, for $1 \leq i \leq M_1$, is further partitioned into $M_{2,i}$ cells

of the magnitude quantizer for Q_2. Let us denote by $\mathbf{s}_i \triangleq (s_{i,0}, s_{i,1}, \cdots, s_{i,M_{2,i}}) \in \mathcal{S}_{M_{2,i}+1}(r_{i-1}, r_i)$, the ascending vector of thresholds for this refined partition. We will use the notation $C_{i,j} = [s_{i,j-1}, s_{i,j})$ for $1 \leq i \leq M_1$, and $1 \leq j \leq M_{2,i}$[1]. Let us also denote by $\bar{\mathbf{s}}$ the M_1-tuple $(\mathbf{s}_1, \cdots, \mathbf{s}_{M_1})$, and by $\mathbf{M}_2$ the M_1-tuple $(M_{2,1}, \cdots, M_{2,M_1})$. Additionally, the fact that Q_2 is a refinement of Q_1 implies that the number of phase regions of the phase quantizer corresponding to magnitude level $C_{i,j}$, denoted by $\tilde{P}_{i,j}$, is a multiple of P_i, i.e., $\tilde{P}_{i,j} = P_i P_{i,j}$, for some $P_{i,j} \in \mathbb{Z}_+$, where $\mathbb{Z}_+$ denotes the set of positive integers. Further, let us denote by $\mathbf{P}_i \triangleq \{P_{i,1}, P_{i,2}, \cdots, P_{i,M_{2,i}}\}$, and by $\bar{\mathbf{P}}$ the M_1-tuple $(\mathbf{P}_1, \cdots, \mathbf{P}_{M_1})$. Accordingly, each quantization bin of the UPQ Q_2 in the EC-SRUPQ case can be represented as

$$\mathcal{R}(i,j,k') = \left\{ re^{j\theta} | s_{i,j-1} \leq r < s_{i,j}, (k'-1)\frac{2\pi}{\tilde{P}_{i,j}} \leq \theta < k'\frac{2\pi}{\tilde{P}_{i,j}} \right\},$$

for $1 \leq i \leq M_1$, $1 \leq j \leq M_{2,i}$ and $1 \leq k' \leq \tilde{P}_{i,j}$. The total number of quantization bins of Q_2 is then $N(Q_2) = \sum_{i=1}^{M_1} \sum_{j=1}^{M_{2,i}} \tilde{P}_{i,j}$. The optimal reconstructed magnitude-phase pair, for each $1 \leq i \leq M_1$, $1 \leq j \leq M_{2,i}$ and $1 \leq k' \leq \tilde{P}_{i,j}$ is $A_{i,j}e^{j\theta_{i,j,k'}}$ given by

$$\theta_{i,j,k'} = (2k'-1)\pi/(\tilde{P}_{i,j}), \tag{3.1}$$

$$A_{i,j} = \text{sinc}\left(\tfrac{1}{\tilde{P}_{i,j}}\right) x(C_{i,j}), \tag{3.2}$$

where $x(C) = \frac{\int_C r g(r) dr}{\int_C g(r) dr}$.

We will use the squared error as a distortion measure. Therefore, the expected

[1]In the entropy-constrained case, the number of magnitude levels could be infinite for both coarse and fine UPQs. However, we consider finite number of magnitude levels, as in prior work.

distortion (per sample) of Q_1 and Q_2 can be expressed, respectively, as

$$D(Q_1) = \frac{1}{2}\left(\int_0^\infty r^2 g(r)dr - \sum_{i=1}^{M_1} A_i^2 q(C_i)\right), \tag{3.3}$$

$$D(Q_2) = \frac{1}{2}\left(\int_0^\infty r^2 g(r)dr - \sum_{i=1}^{M_1}\sum_{j=1}^{M_{2,i}} A_{i,j}^2 q(C_{i,j})\right), \tag{3.4}$$

where for $C \subseteq \mathbb{R}$, $q(C) = \int_C g(r)dr$.

Notice that the tuples $\mathbf{r}$, $\mathbf{P}$, $\bar{\mathbf{s}}$ and $\bar{\mathbf{P}}$ completely specify the SRUPQ.

Let $R(Q_1)$ and $R(Q_2)$ denote the rate of Q_1 and Q_2, respectively. In the EC-SRUPQ case, the rates (in bits/sample) can be expressed as

$$R(Q_1) = \frac{1}{2}\sum_{i=1}^{M_1} q(C_i)(-\log_2 q(C_i) + \log_2 P_i), \tag{3.5}$$

$$R(Q_2) = \frac{1}{2}\sum_{i=1}^{M_1}\sum_{j=1}^{M_{2,i}} q(C_{i,j})\left(-\log_2 q(C_{i,j}) + \log_2(P_i P_{i,j})\right). \tag{3.6}$$

In the FR case we have

$$R(Q_1) = \frac{1}{2}\log_2 N(Q_1), \quad R(Q_2) = \frac{1}{2}\log_2 N(Q_2). \tag{3.7}$$

We will further assume that the thresholds of the magnitude quantizers of UPQs Q_1 and Q_2 take values in some predefined finite sets $\mathcal{A} = \{a_1, a_2, \cdots, a_{K_1}\}$ and $\mathcal{B} = \{b_1, b_2, \cdots, b_{K_2}\}$, respectively. Note that the set $\mathcal{B}$ is finer than $\mathcal{A}$, i.e., $\mathcal{A} \subset \mathcal{B}$, since $\mathcal{A}$ is for the coarse quantizer. In practice, these sets can be obtained by finely discretizing the interval $[0, B]$, for some B chosen such that the probability that $r \notin [0, B]$, to be sufficiently small. Assume that the elements of $\mathcal{A}$ and $\mathcal{B}$ are labeled in increasing order, i.e., $a_i < a_{i+1}$, for $1 \leq i \leq K_1 - 1$, and $b_j < b_{j+1}$, for $1 \leq j \leq K_2 - 1$.

Additionally, let us denote $a_0 = b_0 = 0$, $a_{K_1+1} = b_{K_2+1} = \infty$. Since $\mathcal{A} \subseteq \mathcal{B}$ it follows that there is an injective mapping $\nu : \{0, 1, \cdots, K_1 + 1\} \to \{0, 1, \cdots, K_2 + 1\}$ such that $a_i = b_{\nu(j)}$.

3.2 Optimal EC-SRUPQ Design Algorithm

3.2.1 Problem Formulation

We formulate the problem of EC-SRUPQ design as the minimization of a weighted sum of distortions and entropies, therefore the cost is

$$\mathcal{L}_{EC}(\mathbf{Q}) \triangleq \rho D(Q_1) + (1-\rho) D(Q_2) + \lambda_1 R(Q_1) + \lambda_2 R(Q_2), \tag{3.8}$$

for some fixed $0 < \rho < 1$ and $\lambda_1, \lambda_2 > 0$. Let us denote by $\mathbf{\Omega}(\mathcal{A}, \mathcal{B})$ the set of all EC-SRUPQs such that the thresholds r_i are from the set $\mathcal{A}$ and the thresholds $s_{i,j}$ are from the set $\mathcal{B}$. Then we consider the following optimization problem

$$\min_{\mathbf{Q} \in \mathbf{\Omega}(\mathcal{A}, \mathcal{B})} \mathcal{L}_{EC}(\mathbf{Q}). \tag{3.9}$$

It is known (Everett III, 1963; Luenberger, 1997) that any EC-SRUPQ $\mathbf{Q} \in \mathbf{\Omega}(\mathcal{A}, \mathcal{B})$ for which the quadruple $(R(Q_1), R(Q_2), D(Q_1), D(Q_2))$ lies on the lower boundary of the convex hull of the set of all such quadruples is a solution to problem (3.9) for some choice of ρ, λ_1 and λ_2.

3.2.2 Major Steps of Solution Algorithm

Notice that the first terms in (3.3) and (3.4) are both constant, therefore we can remove them from the cost function. Further, by taking into account the relations

(2.3), (3.2), (3.5) and (3.6), then problem (3.9) becomes equivalent to minimizing $\mathcal{F}_{EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}, \bar{\mathbf{P}})$, where

$$\mathcal{F}_{EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}, \bar{\mathbf{P}}) = \frac{1}{2}\sum_{i=1}^{M_1}\left(\underbrace{q(C_i)\left(-\rho\,\mathrm{sinc}^2\left(\frac{1}{P_i}\right)x^2(C_i) - \lambda_1 \log_2 q(C_i) + (\lambda_1+\lambda_2)\log_2 P_i\right)}_{\varphi(C_i,P_i)} + \sum_{j=1}^{M_{2,i}} \underbrace{q(C_{i,j})\left(-(1-\rho)\,\mathrm{sinc}^2\left(\frac{1}{P_i P_{i,j}}\right)x^2(C_{i,j}) + \lambda_2\left(-\log_2 q(C_{i,j}) + \log_2 P_{i,j}\right)\right)}_{\eta(C_{i,j},P_i,P_{i,j})}\right). \tag{3.10}$$

By examining the cost function $\mathcal{F}_{EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}, \bar{\mathbf{P}})$ we notice that for each pair (i, j) the variable $P_{i,j}$ appears only in the term $\eta(C_{i,j}, P_i, P_{i,j})$. Thus, $P_{i,j}$ can be optimized separately for fixed $C_{i,j}$ and P_i. For $C \subseteq \mathbb{R}$ and positive integer P denote

$$P^*_{C,P} = \arg\min_{P' \in \mathbb{Z}_+} \eta(C, P, P'), \tag{3.11}$$

$$\eta^*(C, P) = \eta(C, P, P^*_{C,P}). \tag{3.12}$$

Note that if there are more minimizers in (3.11), the smallest one is taken. The fact that the minimum in (3.11) exists will be explained in the following section.

Now replace in $\mathcal{F}_{EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}, \bar{\mathbf{P}})$ $P_{i,j}$ by $P^*_{C_{i,j},P_i}$, for each $1 \le i \le M_1$ and $1 \le j \le M_{2,i}$, and denote by $\mathcal{F}_{1,EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}})$ the expression obtained. In other words,

$$\mathcal{F}_{1,EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}) = \frac{1}{2}\sum_{i=1}^{M_1}\left(\varphi(C_i, P_i) + \sum_{j=1}^{M_{2,i}} \eta^*(C_{i,j}, P_i)\right). \tag{3.13}$$

Since $\mathcal{F}_{EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}, \bar{\mathbf{P}}) \ge \mathcal{F}_{1,EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}})$, problem (3.9) can be reduced to minimizing $\mathcal{F}_{1,EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}})$. The expression of $\mathcal{F}_{1,EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}})$ indicates that if the values $\eta^*(C_{i,j}, P_i)$

are known for each possible pair $(C_{i,j}, P_i)$, then the partition of C_i into cells $C_{i,j}$ can be optimized separately for each pair (C_i, P_i). We will denote by $\mathbf{s}^*(C_i, P_i)$ this optimal partition. More generally, for each $C \subseteq \mathbb{R}$ such that C is an interval with endpoints α and β, $\alpha < \beta$, and positive integer P denote

$$\mathbf{s}^*(C, P) = \arg \min_{M, \mathbf{s} \in \mathcal{S}_{M+1}(\alpha,\beta) \cap \mathcal{B}^{M+1}} \sum_{j=1}^{M} \eta^*(C_j, P), \tag{3.14}$$

where $\mathbf{s} = (s_0, \cdots, s_M)$ and $C_j = [s_{j-1}, s_j)$ for $1 \leq j \leq M$. Further, let

$$\gamma^*(C, P) = \sum_{j=1}^{M^*} \eta^*(C_j^*, P) \tag{3.15}$$

where $\mathbf{s}^*(C, P) = (s_0^*, \cdots, s_{M^*}^*)$ and $C_j^* = [s_{j-1}^*, s_j^*)$ for $1 \leq j \leq M^*$. By replacing in $\mathcal{F}_{1,EC}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}})$ each $\mathbf{s}_i$ by the optimal partition $\mathbf{s}^*(C_i, P_i)$, the cost becomes only a function of $\mathbf{r}$ and $\mathbf{P}$ and we denote it by $\mathcal{F}_{2,EC}(\mathbf{r}, \mathbf{P})$. More specifically,

$$\mathcal{F}_{2,EC}(\mathbf{r}, \mathbf{P}) = \frac{1}{2} \sum_{i=1}^{M_1} \left(\varphi(C_i, P_i) + \gamma^*(C_i, P_i) \right). \tag{3.16}$$

Now it can be seen that if the values $\gamma^*(C_i, P_i)$ are known for all possible pairs (C_i, P_i), then the optimal P_i can be found independently for each C_i. Denote for each $C \subseteq \mathbb{R}$

$$P_C^* = \arg \min_{P \in \mathbb{Z}_+} \left(\varphi(C, P) + \gamma^*(C, P) \right), \tag{3.17}$$

where the smallest one is taken if there are multiple minimizers in (3.17). The proof of the fact that the minimum in (3.17) exists follows the same lines as the proof of Proposition 3.4.

By replacing P_i in $\mathcal{F}_{2,EC}(\mathbf{r}, \mathbf{P})$ with $P_{C_i}^*$, we obtain a new cost function which only

depends on $\mathbf{r}$,

$$\mathcal{F}_{3,EC}(\mathbf{r}) = \frac{1}{2}\sum_{i=1}^{M_1}\left(\varphi(C_i, P^*_{C_i}) + \gamma^*(C_i, P^*_{C_i})\right). \tag{3.18}$$

Thus the optimization problem reduces to

$$\begin{aligned} &\min_{M_1,\mathbf{r}} \mathcal{F}_{3,EC}(\mathbf{r}) \\ &\text{subject to} \quad r_i \in \mathcal{A}, \quad 1 \le i \le M_1 - 1. \end{aligned} \tag{3.19}$$

The above discussion suggests the following procedure to solve problem (3.9).

Step 1) For each pair (b_m, b_n), $0 \le m < n \le K_2 + 1$, and each positive integer $P \le P_{cmax}$, compute $P^*_{[b_m,b_n),P}$ defined in (3.11).

Step 2) For each pair (a_u, a_v), $0 \le u < v \le K_1 + 1$, and each positive integer $P \le P_{cmax}$, compute the best partition $\mathbf{s}^*([a_u, a_v), P)$ defined in (3.14).

Step 3) For each pair (a_u, a_v), $0 \le u < v \le K_1+1$, compute $P^*_{[a_u,a_v)}$ defined in (3.17).

Step 4) Solve problem (3.19).

3.2.3 Solution for Each Step

Next we present the details for solving each step, starting with Step 1. For any $y > 0$, denote $f(y) = -\text{sinc}^2(\frac{1}{y})$ and $g(y) = \ln y$ and consider the following minimization problem

$$\min_{P' \in \mathbb{Z}_+} (f(PP') + \delta g(PP')), \tag{3.20}$$

where $\delta > 0$. As P is fixed, we point out that the optimal solution to (3.20) will not be changed by using $g(PP')$ instead of $g(P')$, since $g(PP') = g(P) + g(P')$. Then it can be easily verified that $P^*_{[b_m,b_n),P}$ is the optimal solution to problem (3.20) for $\delta = \frac{\lambda_2}{(1-\rho)x([b_m,b_n))^2 \ln 2}$.

For any positive integer m, let $S(m)$ denote the point in the plane having coordinates $(g(m), f(m))$. Additionally, let $\mathcal{U} \triangleq \{S(m)|m \geq 1\}$ and $\mathcal{U}_P \triangleq \{S(PP')|P' \geq 1\}$. Further, let $\hat{\mathcal{P}}$ denote the set of integers m such that $S(m)$ is on the lower boundary of the convex hull of $\mathcal{U}$. Additionally, let $\hat{\mathcal{P}}_P$ denote the set of integers P' such that $S(PP')$ is on the lower boundary of the convex hull of $\mathcal{U}_P$. It is known (Everett III, 1963; Luenberger, 1997) that some value P'^* minimizes the cost in (3.20) if and only if the point $P'^* \in \hat{\mathcal{P}}_P$ and the line of slope $-\delta$ passing through $S(PP'^*)$ is a support line for $\mathcal{U}_P$. The latter condition is equivalent to

$$left\_slope_P(P'^*) \leq -\delta \leq right\_slope_P(P'^*),$$

where $left\_slope_P(P'^*)$ (respectively, $right\_slope_P(P'^*)$) denotes the slope of the convex hull edge to the left (respectively, right) of $S(PP'^*)$, except for the first and the last ones. Note that for any $\delta > 0$, there is an integer P'^* achieving the minimum in (3.20). The proof is very similar to the proof of Lemma 2.1 in Section 2.2.3. The following proposition then characterizes the set $\hat{\mathcal{P}}_P$.

Proposition 3.1. $\hat{\mathcal{P}}_P = \begin{cases} \mathbb{Z}_+ \setminus \{2\}, & \text{if } P = 1, \\ \mathbb{Z}_+, & \text{if } P \geq 2. \end{cases}$

Proof:

It was proved in appendix B that $\hat{\mathcal{P}} = \mathbb{Z}_+ \setminus \{2\}$. Clearly, when $P = 1$, $\hat{\mathcal{P}}_P = \hat{\mathcal{P}}$, thus the claim holds. Now consider the case $P \geq 2$. If $PP' \geq 3$ then $PP' \in \hat{\mathcal{P}}$, therefore $P' \in \hat{\mathcal{P}}_P$. This implies that for $P \geq 3$ we have $\hat{\mathcal{P}}_P = \mathbb{Z}_+$, while for $P = 2$ we have $\mathbb{Z}_+ \setminus \{1\} \subseteq \hat{\mathcal{P}}_P$. The fact that $1 \in \hat{\mathcal{P}}_2$ can be verified easily concluding the proof.

The monotonicity property established by the following result will be exploited when computing the value $P^*_{[b_m, b_n), P}$, where the proof is similar to the proof of Proposition 2.2 in Chapter 2.

Proposition 3.2. For any integers m, m', n, n' such that $0 \leq m < n \leq K_2 + 1$, $0 \leq m' < n' \leq K_2 + 1$, $m \leq m'$ and $n \leq n'$, and for any $P \in \mathbb{Z}_+$ the following inequality holds

$$P^*_{[b_m, b_n), P} \leq P^*_{[b_{m'}, b_{n'}), P}. \tag{3.21}$$

As a consequence, Algorithm 1 in Chapter 2 can be utilized to determine all values $P^*_{[b_m, b_n), P}$, for fixed P, in $O(K_2 P'_{P,max} + K_2^2)$ time, where $P'_{P,max} = P^*_{[b_{K_2}, b_{K_2+1}), P}$ is the maximum of $P^*_{[b_m, b_n), P}$ over all intervals $[b_m, b_n)$, in virtue of Proposition 3.2. Performing this for all $P, 1 \leq P \leq P_{cmax}$, amounts to $O(K_2 \sum_{P=1}^{P_{cmax}} P'_{P,max} + K_2^2 P_{cmax})$ operations. In order to find a closed form for the expression of the running time, the following result will be useful. Its proof is deferred to appendix C.

Proposition 3.3. For each integer $P \geq 2$ the following holds

$$P'_{P,max} \leq \frac{P'_{1,max}}{P} + 1. \tag{3.22}$$

The following proposition clarifies how to compute P_{cmax}. Its proof is deferred to appendix C.

Proposition 3.4. Consider $P_{cmax} = \max\{P'_{1,max} + 1, P''\}$, where P'' is the solution to problem (3.20) for $P = 1$ and $\delta = \frac{\lambda_1}{\rho x ([b_{K_2}, b_{K_2+1})^2 \ln 2}$. Then there is an optimal EC-SRUPQ such that the phase quantizer corresponding to any magnitude level of the coarse UPQ has no more than P_{cmax} levels.

Using P_{cmax} defined in Proposition 3.4 and based on Proposition 3.3 one obtains

$$\sum_{P=1}^{P_{cmax}} P'_{P,max} \leq P'_{1,max} \sum_{P=1}^{P_{cmax}} \frac{1}{P} + P_{cmax} \leq P'_{1,max}(\ln P_{cmax} + 1) + P_{cmax} \leq P_{cmax}(\ln P_{cmax} + 2),$$

where the second inequality follows from the partial sum of Harmonic series, i.e., $\sum_{P=1}^{P_{cmax}} \frac{1}{P} \leq 1 + \int_1^{P_{cmax}} \frac{1}{P}\, dP = \ln P_{max} + 1$. Thus, the running time of Step 1 becomes

$O(K_2 P_{cmax}(\ln P_{cmax} + K_2))$. If $\ln P_{cmax} < K_2$, which is the case in our experiments, the time complexity amounts to $O(K_2^2 P_{cmax})$.

Then we will consider the problem at Step 4. We will show that it is equivalent to an MWP problem in the WDAG $G = (V, E, w)$ where $V = \{0, 1, \cdots, K_1 + 1\}$ is the vertex set, and $E = \{(u, v) \in V^2 \mid 0 \leq u < v \leq K_1 + 1\}$ denotes the edge set. The weight of each edge $(u, v) \in E$ is $w(u, v)$ defined as

$$w(u, v) = \varphi([a_u, a_v), P^*_{[a_u, a_v)}) + \gamma^*([a_u, a_v), P^*_{[a_u, a_v)}). \tag{3.23}$$

Then each ascending vector of thresholds $\mathbf{r} \in \mathcal{S}_{M_1+1}(0, \infty)$, with components in $\mathcal{A}$, is in a one-to-one correspondence with an M_1-edge path in G, from the source node 0 to the final node $K_1 + 1$. It can be easily seen that the weight of the path equals $\mathcal{F}_{3,EC}(\mathbf{r})$. This observation implies that problem (3.19) is equivalent to the MWP problem in the WDAG G. If each edge weight can be evaluated in constant time, this problem can be solved in $O(|V| + |E|) = O(K_1^2)$ operations.

Next we will consider the problem at Step 2. We will show that for each P, the problem can be solved by solving the single source MWP problem in another WDAG, multiple times. For each positive integer P, construct the WDAG $G_P = (V, E, w_P)$, where $V = \{0, 1, \cdots, K_2 + 1\}$ is the vertex set, and $E = \{(m, n) | 0 \leq m < n \leq K_2 + 1\}$ is the edge set. For each edge (m, n) define the weight $w_P(m, n)$ as follows

$$w_P(m, n) \triangleq \eta^*([b_m, b_n), P).$$

Let us fix an arbitrary pair (u, v). Let $C = [a_u, a_v)$. Recall that $[a_u, a_v) = [b_{\nu(u)}, b_{\nu(v)})$. Consider a partition $\mathbf{s}$ of C consisting of M cells. It corresponds to an M-edge path from node $\nu(u)$ to $\nu(v)$ in G_P, and the mapping is one to one. It can be easily

seen that the weight of the path equals $\sum_{j=1}^{M} \eta^*(C_j, P)$. Therefore, finding $\mathbf{s}^*(C, P)$ is equivalent to finding the MWP path from $\nu(u)$ to $\nu(v)$. Since we need to find the MWP from $\nu(u)$ to $\nu(v)$ for any $0 \leq u < v \leq K_1 + 1$, we will solve the single source MWP problem corresponding to source $\nu(u)$. This is the problem of finding the MWP from the source to any other graph node reachable from the source and can be solved $O(|V| + |E|) = O(K_2^2)$. Doing so for each u and P amounts to $O(K_1 K_2^2 P_{cmax})$ operations.

The problem at Step 3 is straightforward and can be solved in $O(K_1^2 P_{cmax})$ operations. Additionally, the cumulative probabilities, the first and second order moments of set $\mathcal{B}$ are precomputed and stored in a preprocessing step as in Section 2.2.3, as set $\mathcal{B}$ is much finer, and this requires only $O(K_2)$ operations. Then each $x[b_m, b_n)$, $x[a_u, a_v)$, $q[b_m, b_n)$ and $q[a_u, a_v)$ can be evaluated in constant time.

In conclusion, problem (3.9) can be solved in $O(K_2 P_{cmax}(K_1 K_2 + \ln P_{cmax} + K_2))$ time. Note that if $\ln P_{cmax} + K_2 \leq K_1 K_2$, which we found to be true in our experiments, then the time complexity of the solution is $O(K_1 K_2^2 P_{cmax})$.

Finally, the following pseudocode in Algorithm 4 finalizes the solution algorithm to problem (3.9). We point out that for each P, $W(m, n, P)$ denotes the minimum weight of the MWP from node m to node n, and $\varepsilon(m, n, P)$ records the intermediate node passed by the MWP. Moreover, for the coarse UPQ, we denote by $\hat{W}(v)$ the weight of the MWP from the source node 0 to node v, and $\epsilon(v)$ records the node preceding v on this optimal path. At the end, the MWP of the coarse UPQ and the fine UPQ can be tracked back by utilizing the values of $\epsilon(v)$ and $\varepsilon(\nu(u), \nu(v), P^*_{[a_u, a_v)})$.

Algorithm 4: Solution algorithm for problem (3.9).

Preprocessing Stage
begin

/ Step 1) */*
for $P = 1$ **to** P_{cmax} **do**
  for $n = 1$ **to** $K_2 + 1$ **do**
    $P^*_{[b_{n-1},b_n),P} := \min \arg\min_{P' \in \hat{\mathcal{P}}_P} \eta([b_{n-1}, b_n), P, P')$
    $W(n-1, n, P) := w_P(n-1, n)$
    $\varepsilon(n-1, n, P) := n-1$
    for $m = n-2$ **down to** 0 **do**
      $P^*_{[b_m,b_n),P} := \min \arg\min_{P^*_{[b_m,b_{n-1}),P} \le P' \le P^*_{[b_{m+1},b_n),P}} \eta([b_m, b_n), P, P')$
      $W(m, n, P) := w_P(m, n)$
      $\varepsilon(m, n, P) := m$

/ Step 2) */*
for $P = 1$ **to** P_{cmax} **do**
  for $u = 0$ **to** K_1 **do**
    for $v = u + 1$ **to** $K_1 + 1$ **do**
      for $k = \nu(u) + 1$ **to** $\nu(v) - 1$ **do**
        if $W(\nu(u), \nu(v), P) > W(\nu(u), k, P) + W(k, \nu(v), P)$ **then**
          $W(\nu(u), \nu(v), P) := W(\nu(u), k, P) + W(k, \nu(v), P)$
          $\varepsilon(\nu(u), \nu(v), P) := k$

/ Step 3) and Step 4) */*
$\hat{W}(0) = 0$
for $u = 0$ **to** K_1 **do**
  for $v = u + 1$ **to** $K_1 + 1$ **do**
    for $P = 1$ **to** P_{cmax} **do**
      Evaluate $\gamma^*([a_u, a_v), P)$ using (3.15)
      Evaluate $P^*_{[a_u,a_v)}$ using (3.17)
    if $\left(\hat{W}(u) + w(u, v) < \hat{W}(v)\right)$ **then**
      $\hat{W}(v) := \hat{W}(u) + w(u, v)$
      $\epsilon(v) := u$

Restore the vectors $\mathbf{r}$ and $\mathbf{P}$ corresponding to the coarse UPQ Q_1
Restore the vectors $\bar{\mathbf{s}}$ and $\bar{\mathbf{P}}$ corresponding to the fine UPQ Q_2

3.3 Optimal FR-SRUPQ Design Algorithm

3.3.1 Problem Formulation

In the fixed-rate case, according to (3.7), the rates $R(Q_1)$ and $R(Q_2)$ are determined by the number of quantization cells $N(Q_1)$ and $N(Q_2)$, respectively. Therefore, the problem of FR-SRUPQ design can be formulated as the constrained problem of minimizing a weighted sum of the distortions with constraints on the number of quantizer levels, i.e.,

$$\begin{aligned} &\min_{M_1,\mathbf{r},\mathbf{P},\bar{\mathbf{s}},\bar{\mathbf{P}}} \quad \rho D(Q_1) + (1-\rho) D(Q_2) \\ &\text{subject to} \quad \sum_{i=1}^{M_1} P_i = N_1, \quad \sum_{j=1}^{M_{2,i}} P_{i,j} = N' \\ &r_i \in \mathcal{A}, \quad s_{i,j} \in \mathcal{B}, \quad 1 \le i \le M_1 - 1, \quad 1 \le j \le M_2 - 1, \end{aligned} \tag{3.24}$$

where N_1 and $N_2 = N_1 N'$ are the two target numbers of quantization cells of Q_1 and Q_2, respectively. Moreover, $R_i = \lceil \log_2 N_i \rceil / 2$, $i = 1, 2$, denote the desired rates (bits/sample) of UPQs Q_1 and Q_2. Imposing the constraint on $\sum_{j=1}^{M_{2,i}} P_{i,j}$ is motivated by the fact that the value $\log_2 N' = \log_2 \frac{N_2}{N_1}$ is actually the amount of extra bits appended to each binary index output by the coarse quantizer Q_1 to obtain an index of the fine quantizer Q_2.

3.3.2 Solution Algorithm

Since the first terms in (3.3) and (3.4) are both constant, then problem (3.24) is equivalent to minimizing $\mathcal{F}_{FR}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}, \bar{\mathbf{P}})$, where

$$\begin{aligned}\mathcal{F}_{FR}(\mathbf{r}, \mathbf{P}, \bar{\mathbf{s}}, \bar{\mathbf{P}}) = \frac{1}{2}\sum_{i=1}^{M_1}\Bigg(&\underbrace{-q(C_i)\rho\ \mathrm{sinc}^2\left(\frac{1}{P_i}\right)x^2(C_i)}_{\varphi'(C_i,P_i)} + \\ &(1-\rho)\underbrace{\sum_{j=1}^{M_{2,i}}\left(-q(C_{i,j})\ \mathrm{sinc}^2\left(\frac{1}{P_iP_{i,j}}\right)x^2(C_{i,j})\right)}_{\xi(C_i,P_i,\mathbf{s}_i,\mathbf{P}_i)}\Bigg).\end{aligned} \tag{3.25}$$

It is noticed from the above cost function that $\xi(C_i, P_i, \mathbf{s}_i, \mathbf{P}_i)$ can be optimized separately for fixed C_i and P_i. Then for $C_i \subseteq \mathbb{R}$ and positive integer P_i denote

$$\xi^*(C_i, P_i) \triangleq \min_{\substack{M_{2,i}, \mathbf{s}_i \in \mathcal{S}_{M_{2,i}+1}(C_i)\cap\mathcal{B}^{M_{2,i}+1}, \\ \mathbf{P}_i, \sum_{j=1}^{M_{2,i}} P_{i,j} = N'}} \xi(C_i, P_i, \mathbf{s}_i, \mathbf{P}_i). \tag{3.26}$$

Further, let $\mathbf{s}^*(C_i, P_i)$ denote the partition $\mathbf{s}_i$ achieving the minimum in (3.26) and let $\mathbf{P}^*(C_i, P_i)$ be the corresponding optimal $M_{2,i}$-tuple $\mathbf{P}_i$. Then problem (3.24) reduces to solving

$$\begin{aligned}\min_{M_1, \mathbf{r}, \mathbf{P}} \mathcal{F}_{1,FR}(\mathbf{r}, \mathbf{P}) &= \frac{1}{2}\sum_{i=1}^{M_1}\left(\varphi'(C_i, P_i) + (1-\rho)\xi^*(C_i, P_i)\right) \\ \text{subject to } \sum_{i=1}^{M_1} P_i = N_1, &\quad r_i \in \mathcal{A}, \quad 1 \le i \le M_1 - 1.\end{aligned} \tag{3.27}$$

We conclude that the solution to problem (3.24) can be broken into two steps as follows.

Step 1) For each pair (a_u, a_v), $0 \le u < v \le K_1 + 1$, and each positive integer $P \le N_1$,

compute $\xi^*([a_u, a_v), P)$, $\mathbf{s}^*([a_u, a_v), P)$ and $\mathbf{P}^*([a_u, a_v), P)$ by solving (3.26).

Step 2) Solve problem (3.27).

Next we will deal with the problem at the first step. For each positive integers P and P' and for $b_0 \leq \alpha < \beta \leq b_{K_2+1}$ denote

$$\omega_{P,P'}(\alpha, \beta) \triangleq -\text{sinc}^2\left(\frac{1}{PP'}\right)(x[\alpha, \beta))^2 \, q[\alpha, \beta). \tag{3.28}$$

For each positive integer P, each partition $\mathbf{s} = (s_0, s_1, \cdots, s_M)$ of M cells and each M-tuple of positive integers $\mathbf{P} = (P'_1, \cdots, P'_M)$ denote

$$\mathcal{O}_P(\mathbf{s}, \mathbf{P}) \triangleq \sum_{j=1}^{M} \omega_{P,P'_j}(s_{j-1}, s_j). \tag{3.29}$$

Then it is noted from (3.26) that for each pair (u, v) the following holds

$$\xi^*([a_u, a_v), P) = \min_{M, \mathbf{s} \in \mathcal{S}_{M+1}([a_u, a_v)) \cap \mathcal{B}^{M+1}, \mathbf{P} \in \mathbb{Z}_+^M, \sum_{j=1}^{M} P'_j = N'} \mathcal{O}_P(\mathbf{s}, \mathbf{P}), \tag{3.30}$$

where $\mathbf{P} = (P'_1, \cdots, P'_M)$. This problem is very similar to the optimal FRUPQ design problem treated in Section 2.3. Therefore, we can use the same solution algorithm. More specifically, for each u we use the algorithm in Section 2.3.3 to solve problem (3.30) for the pair $(u, v) = (u, K_2 + 1)$. The algorithm runs in $O(K_2 N'^2)$ time and also solves the problem for all pairs (u, v) where $u < v < K_2 + 1$. This procedure is repeated for each u and then the whole process is repeated for each P. Thus, the total time complexity amounts to $O(K_1 K_2 N'^2 N_1)$.

Let us give some details. For each u, P and each pair of positive integers (k, n)

with $\nu(u) \leq k \leq K_2 + 1$ and $1 \leq n \leq N'$, consider the problem $\mathcal{P}_{P,u}(k,n)$ defined as

$$\begin{aligned} &\min_{M,\mathbf{s}\in\mathcal{S}_{M+1}([b_{\nu(u)},b_k))\cap\mathcal{B}^{M+1},\mathbf{P}\in\mathbb{Z}_+^M} \quad \mathcal{O}_P(\mathbf{s},\mathbf{P}) \\ &\text{subject to } \sum_{j=1}^{M} P'_j = n. \end{aligned} \tag{3.31}$$

Additionally, denote by $\hat{\mathcal{O}}_{P,u}(k,n)$ the optimal value of the objective function in (3.31). Notice that problem (3.31) is similar to problem $\mathcal{P}(k,n)$ in Section 2.3. It can be solved using the following recurrence relation

$$\hat{\mathcal{O}}_{P,u}(k,n) = \min_{0\leq t<n} \min_{\nu(u)\leq j<k} \left(\hat{\mathcal{O}}_{P,u}(j,t) + \omega_{P,n-t}(b_j,b_k)\right), \tag{3.32}$$

where $\hat{\mathcal{O}}_{P,u}(\nu(u),0) = 0$ and $\hat{\mathcal{O}}_{P,u}(\nu(u),t) = \hat{\mathcal{O}}_{P,u}(j,0) = \infty$, for $t > 0$ and $j \geq 1$.

We point out that for fixed P and u, the straightforward solution to problem (3.32) will take $O(K_2^2 N'^2)$ time, as there are $O(K_2 N')$ pairs (k,n) in total. On the other hand, note that the Monge property holds for the cost function in problem (3.32), by following Section 2.3.3, for fixed P and u. Specifically, the Monge property can be utilized to solve the minimization over j in (3.32). This reduces the complexity of solving (3.32) to $O(K_2 N'^2)$, and further leads to the aforementioned $O(K_1 K_2 N'^2 N_1)$ time complexity of Step 1.

Subsequently, Step 1 can be completed by solving the problem $\mathcal{P}_{P,u}(K_2+1, N')$ for each integer $P \leq N_1$ and each pair (a_u, a_v), $0 \leq u < v \leq K_1 + 1$.

Now let us consider the problem at Step 2. For any cell $C = [a_u, a_v)$ and any positive integer P denote

$$\omega'_P(a_u,a_v) \triangleq \frac{1}{2}\left(\rho f(P)\left(x[a_u,a_v)\right)^2 q[a_u,a_v) + (1-\rho)\xi^*([a_u,a_v),P)\right). \tag{3.33}$$

Then problem (3.27) is equivalent to

$$\begin{aligned} \min_{M_1,\mathbf{r}\in\mathcal{S}_{M_1+1}(0,\infty)\cap\mathcal{A}^{M_1+1},\mathbf{P}\in\mathbb{Z}_+^{M_1}} \quad & \mathcal{O}(\mathbf{r},\mathbf{P}) \triangleq \textstyle\sum_{i=1}^{M_1}\omega_{P_i}(r_{i-1},r_i), \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{M_1} P_i = N_1, \end{aligned} \tag{3.34}$$

where $\mathbf{P} = (P_1, \cdots, P_{M_1})$. The above problem is also similar to optimal FRUPQ design problem and can be solved using dynamic programming. However, the weights might not satisfy the Monge property, therefore there is no time complexity reduction. Thus, solving Step 2 will need $O(K_1^2 N_1^2)$ operations. In conclusion, the time complexity for the proposed FR-SRUPQ design is $O(K_1 K_2 N'^2 N_1)$.

3.4 Experimental Results

This section assesses the practical performance of the EC-SRUPQ design algorithm presented in this chapter, and compares it with the theoretical bounds. The experiments are conducted for the same two-dimensional random vector (X_1, X_2) as in Chapter 2, where X_1 and X_2 are i.i.d. Gaussian variables with zero-mean and unit-variance.

The finite sets of possible thresholds $\mathcal{A}$ and $\mathcal{B}$ are obtained by dividing the range $[0, 6]$ into subintervals of size 0.025. In other words, $K_1 = K_2 = 240$ and $a_i = b_i = 0.025i$, for $1 \leq i \leq K_1$. Moreover, we set $P_{cmax} = 55$, $\rho = 0.03, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5$. The value of P'_{max} is 40.

In this section, the notations R_i and D_i are utilized instead of $R(Q_i)$ and $D(Q_i)$, respectively, for $i = 1, 2$. Additionally, let $R(D_i)$ denote the rate-distortion function for the Gaussian source, i.e., $R(D_i) = -0.5\log_2(D_i)$.

Figure 3.1, Figure 3.2 and Figure 3.3 illustrate the performance of the proposed

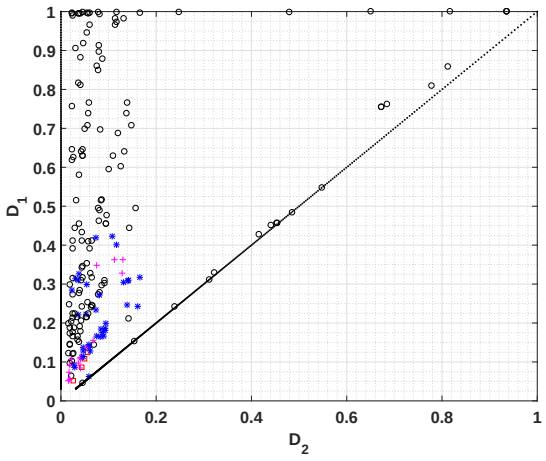


Figure 3.1: Distortion performance of the proposed EC-SRUPQ.

EC-SRUPQ, in terms of distortion pair (D_1, D_2), rate pair (R_1, R_2) and the rate-gap pair $(R_1 - R(D_1), R_2 - R(D_2))$, respectively.

It can be noticed from Figure 3.3 that in most cases the gap $R_2 - R(D_2)$ is within 0.275 bits/sample, which is very close to the gap of 0.254 bits/sample. Note that the existence of this gap is expected since the theoretical bound is achieved using vector quantization with dimension approaching ∞ while we use scalar quantization. The rate gap between the optimum ECSQ and the rate-distortion limit was proved in (Gish and Pierce, 1968) to be $\frac{1}{2}\log_2 \frac{2\pi e}{12} = 0.2546$ bits/sample at high resolution. Most of the points in this category also have the gap in $R_1 - R(D_1)$ within this limit. However, there are also cases in which there is some additional loss only in R_1, only in R_2, or in both. These cases are represented in the three figures with stars, crosses, and squares, respectively. We can see that the cases with extra loss occur mostly when both distortions are small. The existence of such extra loss in rate could be attributed to the additional tension induced in the optimization by competing requirements at the two decoders, as opposed to one decoder.

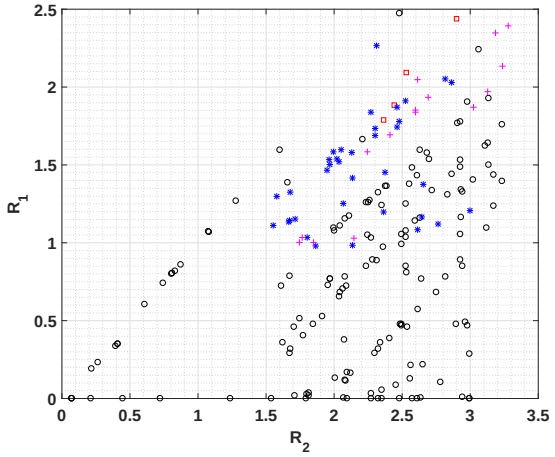


Figure 3.2: Rate performance of the proposed EC-SRUPQ.

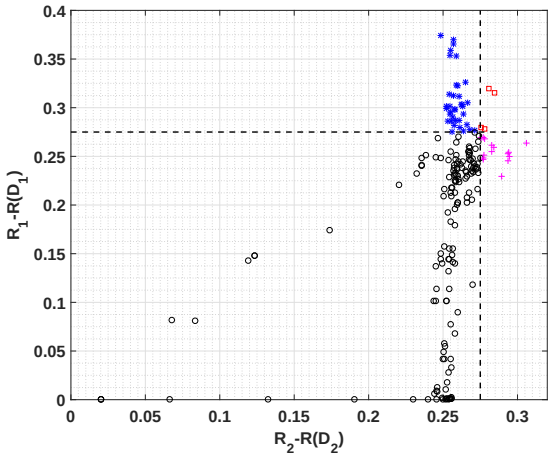


Figure 3.3: Gap in rate versus the theoretical lower bounds.

3.5 Conclusion

This chapter presents the algorithms for globally optimal design of SRUPQ for bivariate circularly symmetric sources, for both the EC and FR cases. The global optimality holds when the magnitude quantizers thresholds are confined to some finite sets. For the EC case, the cost to be minimized is a weighted sum of distortions and entropies, and the proposed algorithm involves a series of stages including solving the MWP problem for multiple node pairs in certain WDAGs. In the FR case, the proposed solution is based on tackling with a series of dynamic programming problems. The experimental results performed on a bivariate circularly symmetric Gaussian source demonstrate the effectiveness in practice of the proposed EC-SRUPQ scheme.

Chapter 4

Design of Scalar Quantizer for Sequential Coding of Correlated Sources

This chapter addresses the design of an SSQ for finite-alphabet correlated sources in the FR and EC cases. The optimization problem is formulated as the minimization of a weighted sum of distortions and rates. The proposed solution is globally optimal for the class of SSQs with convex cells and is based on solving the MWP problem in the EC case, respectively, a length-constrained MWP problem in the FR case, in a series of WDAGs. The asymptotic time complexity is $O(K_1^2 K_2^2)$, where K_1 and K_2 are the respective sizes of the alphabets of the two sources. Additionally, it is proved that, by applying the proposed algorithms to finite, uniform discretizations of correlated sources with continuous joint pdf, the performance approaches that of the optimal EC-SSQ, respectively FR-SSQ, with convex cells for the original sources as the accuracy of the discretization increases. Extensive experiments performed with correlated Gaussian sources validate the effectiveness in practice of the proposed approach in

approximating the optimal SSQ for the case of continuous-alphabet sources.

This chapter is organized as follows. The next section introduces the necessary definitions and notations. Section 4.2 formulates the problem of optimal EC-SSQ design and presents the proposed solution algorithm. The problem of optimal FR-SSQ design and its solution are presented in Section 4.3. Section 4.4 investigates the application of the proposed designs to continuous sources. Section 4.5 shows simulation results, and finally, Section 4.6 concludes the chapter.

4.1 Notations and Problem Formulation

This section presents the definitions and notations used throughout this chapter. Let X and Y be two finite-alphabet jointly distributed random variables (RVs). Let p_{XY} denote their joint pmf. The RVs X and Y take values in the alphabets $\mathcal{X} = \{x_1, \ldots, x_{K_1}\} \subseteq \mathbb{R}$, respectively $\mathcal{Y} = \{y_1, \ldots, y_{K_2}\} \subseteq \mathbb{R}$, where K_1 and K_2 are positive integers, $x_i < x_{i+1}$, for $1 \leq i \leq K_1 - 1$ and $y_j < y_{j+1}$, for $1 \leq j \leq K_2 - 1$. Let p_X and p_Y denote the marginal pmfs of X and Y, respectively.

For any positive integer k denote $I_k \triangleq \{0, \ldots, k\}$ and $E_k \triangleq \{(u, v) | 0 \leq u < v \leq k\}$. For any $(u, v) \in E_{K_1}$ let $C_X(u, v] \triangleq (x_u, x_v] \cap \mathcal{X} = \{x_{u+1}, \ldots, x_v\}$. For any $(m, n) \in E_{K_2}$ denote $C_Y(m, n] \triangleq (y_m, y_n] \cap \mathcal{Y} = \{y_{m+1}, \ldots, y_n\}$. In this chapter we consider quantizers with convex cells[1]. A subset of $\mathcal{X}$ is said to be convex if it equals $C_X(u, v]$ for some $(u, v) \in E_{K_1}$, while any convex subset of $\mathcal{Y}$ equals $C_Y(m, n]$ for some $(m, n) \in E_{K_2}$.

Note that in this chapter, for any positive integer M, an ascending M-sequence for X is defined as a vector of integer thresholds $\mathbf{r} \triangleq (r_0, r_1, \ldots, r_M)$, such that $r_0 = 0 <$

[1]Note that the cell convexity does not preclude the optimality of the quantizers for the source Y (Muresan and Effros, 2008; Gyorgy and Linder, 2002), as mentioned in Section 1.2.3, but there may be some loss of optimality for the quantizer for source X.

$r_1 < \ldots < r_{M-1} < r_M = K_1$. Let us denote by $\mathfrak{T}_X(M)$ the set of all such sequences. Furthermore, let $\mathfrak{T}_X \triangleq \cup_{M>0}\mathfrak{T}_X(M)$. Clearly, the encoder partition of any scalar quantizer with M convex cells for the source X can be identified with the ascending M-sequence $\mathbf{r} \in \mathfrak{T}_X(M)$, where $C_X(r_{i-1}, r_i]$ is the i-th cell, for $1 \leq i \leq M$. Similarly, an ascending M-sequence for Y is any vector of integer thresholds $\mathbf{s} = (s_0, s_1, \ldots, s_M)$ such that $s_0 = 0 < s_1 < \ldots < s_{M-1} < s_M = K_2$. We use the notation $\mathfrak{T}_Y(M)$ for the set of all ascending M-sequences for Y, and $\mathfrak{T}_Y \triangleq \cup_{M>0}\mathfrak{T}_Y(M)$. The encoder partition of any quantizer with M convex cells for the source Y can be identified with the ascending M-sequence $\mathbf{s} \in \mathfrak{T}_Y(M)$, where $C_Y(s_{j-1}, s_j]$ is the j-th cell, for $1 \leq j \leq M$. In the sequel we use interchangeably the terms ascending sequence and quantizer (or encoder) partition.

In this chapter we also use the fact that any quantizer with convex cells can be naturally associated with a path in a certain WDAG. Note that for any positive integer k, the k-edge MWP problem is the problem of finding the path of minimum weight among all paths from the source to the final node which have exactly k edges. This is a length-constrained MWP problem since the number of edges can be regarded as the length of the path.

For any mapping $\eta : E_{K_1} \to \mathbb{R}$, let $G_X(\eta)$ denote the WDAG where I_{K_1} is set of vertices, E_{K_1} is the set of edges, and η is the weighting function. The source node is the vertex 0 and the final node is the vertex K_1. It can be easily seen that any ascending M-sequence $\mathbf{r} \in \mathfrak{T}_X(M)$ can be identified with an M-edge path in $G_X(\eta)$ from the source to the final node, whose i-th edge is (r_{i-1}, r_i). Clearly, this correspondence between M-ascending sequences for X and M-edge paths from 0 to K_1 is one-to-one.

Likewise, for any mapping $\zeta : E_{K_2} \to \mathbb{R}$ let $G_Y(\zeta)$ denote the WDAG with I_{K_2} as the set of vertices, E_{K_2} as the set of edges, and ζ as the weighting function. Then

there is a one-to-one correspondence between the ascending M-sequences for Y and the M-edge paths from 0 to K_2.

An SSQ for the pair of RVs (X, Y) consists of two encoding functions f_1, f_2, and two decoding functions g_1, g_2

$$f_1 : \mathcal{X} \longrightarrow \mathcal{I}_1, \quad f_2 : \mathcal{I}_1 \times \mathcal{Y} \longrightarrow \mathcal{I}_2, \quad g_1 : \mathcal{I}_1 \longrightarrow \hat{\mathcal{X}}, \quad g_2 : \mathcal{I}_1 \times \mathcal{I}_2 \longrightarrow \hat{\mathcal{Y}}, \tag{4.1}$$

where $\mathcal{I}_1 = \{1, 2, \ldots, M_1\}$ and $\mathcal{I}_2 = \{1, 2, \ldots, M_2\}$ for some positive integers M_1, M_2, $\hat{\mathcal{X}} \subseteq \mathbb{R}$ and $\hat{\mathcal{Y}} \subseteq \mathbb{R}$. Notice that the pair (f_1, g_1) represents a scalar quantizer for the source X with M_1 cells. We will use the notation C_i for the cell assigned index i, i.e., $C_i \triangleq f_1^{-1}(i)$, $1 \leq i \leq M_1$. For each $i, 1 \leq i \leq M_1$, the encoder-decoder pair $(f_2(i, \cdot), g_2(i, \cdot))$ represents a scalar quantizer for the source Y. Let $M_{2,i}$ denote its number of quantizer cells. Note that $M_{2,i} \leq M_2$. Additionally, we use the notation $C_{i,j}$ for the j-th cell of this quantizer, i.e., $C_{i,j} \triangleq \{y \in \mathcal{Y} | f_2(i, y) = j\}$.

We will assume that all aforementioned quantizers contain convex cells. Thus, the encoder partition of each such quantizer is specified by some ascending sequence. Let $\mathbf{r} \triangleq (r_0, r_1, \ldots, r_{M_1}) \in \mathcal{T}_X(M_1)$ be the ascending sequence specifying the encoder partition generated by f_1. In other words, we have $C_i = C_X(r_{i-1}, r_i]$ for $1 \leq i \leq M_1$. Further, for each $1 \leq i \leq M_1$, let $\mathbf{s}_i \triangleq (s_{i,0}, s_{i,1}, \ldots, s_{i,M_{2,i}}) \in \mathcal{T}_Y(M_{2,i})$ be the ascending sequence specifying the encoder partition generated by $f_2(i, \cdot)$. Thus, we have $C_{i,j} = C_Y(s_{i,j-1}, s_{i,j}]$ for $1 \leq j \leq M_{2,i}$. We will denote by $\bar{\mathbf{s}}$ the M_1-tuple $(\mathbf{s}_1, \ldots, \mathbf{s}_{M_1})$.

We will use the squared error as a distortion measure. Thus, the expected distortion at decoder 1, respectively 2, is

$$\begin{aligned} D_1(f_1, g_1) &= \mathbb{E}[(X - \hat{X})^2] = \sum_{i=1}^{M_1} \sum_{x \in C_i} (x - g_1(i))^2 p_X(x), \\ D_2(f_1, f_2, g_2) &= \mathbb{E}[(Y - \hat{Y})^2] = \sum_{i=1}^{M_1} \sum_{j=1}^{M_{2,i}} \sum_{y \in C_{i,j}} (y - g_2(i,j))^2 \sum_{x \in C_i} p_{XY}(x,y). \end{aligned} \tag{4.2}$$

It is known that, for fixed encoders, the decoding functions can be optimized to minimize the distortion by setting

$$g_1(i) = \hat{x}(C_i), \quad g_2(i,j) = \hat{y}(C_{i,j}|C_i), \tag{4.3}$$

for $1 \leq i \leq M_1$, $1 \leq j \leq M_{2,i}$, where, for each set $A \subset \mathcal{X}$, and each $B \subset \mathcal{Y}$, we define

$$\begin{aligned} \hat{x}(A) &\triangleq \frac{\sum_{x \in A} x p_X(x)}{\sum_{x \in A} p_X(x)}, \\ \hat{y}(B|A) &\triangleq \frac{\sum_{y \in B} y \mathbb{P}[Y = y | X \in A]}{\sum_{y \in B} \mathbb{P}[Y = y | X \in A]} \\ &= \frac{\sum_{y \in B} y \sum_{x \in A} p_{XY}(x,y)}{\sum_{y \in B} \sum_{x \in A} p_{XY}(x,y)}. \end{aligned}$$

In the sequel we assume optimized decoders. Thus, the distortions at decoder 1, respectively 2, depend only on the encoders, which are completely specified by their partitions. Therefore, we denote them from now on by $D_1(\mathbf{r})$, respectively $D_2(\mathbf{r}, \bar{\mathbf{s}})$. By plugging (4.3) in (4.2) we obtain

$$\begin{aligned} D_1(\mathbf{r}) &= \sum_{i=1}^{M_1} \sum_{x \in C_i} (x - \hat{x}(C_i))^2 p_X(x), \\ D_2(\mathbf{r}, \bar{\mathbf{s}}) &= \sum_{i=1}^{M_1} \sum_{j=1}^{M_{2,i}} \sum_{y \in C_{i,j}} (y - \hat{y}(C_{i,j}|C_i))^2 \sum_{x \in C_i} p_{XY}(x,y). \end{aligned} \tag{4.4}$$

Let $R_1(\mathbf{r})$ denote the rate of encoder 1 and let $R_2(\mathbf{r}, \bar{\mathbf{s}})$ be the rate of encoder 2. The expression of the rates depends on whether the quantizers are FR or EC. Therefore, from now on we will discuss the two cases separately. In the following section we formulate the problem of optimal EC-SSQ design and propose a solution algorithm. The counterpart for the FR case is addressed in Section 4.3.

4.2 Optimal EC-SSQ Design Algorithm

Let I and J be the random variables representing the indexes output by f_1, respectively f_2. In the EC case the rate at encoder 1 equals the entropy of I, while the rate at encoder 2 equals the conditional entropy of J conditioned on I. Thus, we have

$$\begin{aligned} R_1(\mathbf{r}) &= -\sum_{i=1}^{M_1} P(C_i) \log_2 P(C_i), \\ R_2(\mathbf{r}, \bar{\mathbf{s}}) &= -\sum_{i=1}^{M_1} \sum_{j=1}^{M_{2,i}} P(C_i, C_{i,j}) \log_2 P(C_i, C_{i,j}) + \sum_{i=1}^{M_1} P(C_i) \log_2 P(C_i), \end{aligned} \tag{4.5}$$

where $P(C_i) \triangleq \mathbb{P}[X \in C_i]$ and $P(C_i, C_{i,j}) \triangleq \mathbb{P}[X \in C_i, Y \in C_{i,j}]$, for $1 \leq i \leq M_1$ and $1 \leq j \leq M_{2,i}$.

Let $\mathcal{RD}_{EC}$ denote the set of all quadruples $(R_1(\mathbf{r}), R_2(\mathbf{r}, \bar{\mathbf{s}}), D_1(\mathbf{r}), D_2(\mathbf{r}, \bar{\mathbf{s}}))$ for all possible pairs $(\mathbf{r}, \bar{\mathbf{s}})$. Then any point on the lower boundary of the convex hull of $\mathcal{RD}_{EC}$ is optimal in some sense. Any such point is the solution of the minimization of a weighted sum of the distortions and rates $\rho_1 D_1(\mathbf{r}) + \rho_2 D_2(\mathbf{r}, \bar{\mathbf{s}}) + \lambda_1 R_1(\mathbf{r}) + \lambda_2 R_2(\mathbf{r}, \bar{\mathbf{s}})$, for some choice of positive weights ρ_1, ρ_2, λ_1 and λ_2. Notice that the solution of the minimization problem remains the same if all the weights are divided by $\rho_1 + \rho_2$.

Therefore, we formulate the optimization problem as follows

$$\min_{M_1,\mathbf{r}\in\mathcal{T}_X(M_1),\bar{\mathbf{s}}\in\mathcal{T}_Y^{M_1}} \mathcal{F}(\mathbf{r},\bar{\mathbf{s}}) \triangleq \rho D_1(\mathbf{r}) + (1-\rho)D_2(\mathbf{r},\bar{\mathbf{s}}) + \lambda_1 R_1(\mathbf{r}) + \lambda_2 R_2(\mathbf{r},\bar{\mathbf{s}}), \quad (4.6)$$

for some fixed $\rho, 0 < \rho < 1$, $\lambda_1 > 0$ and $\lambda_2 > 0$. We point out that the formulation of the optimization problem as a minimization of a weighted sum of distortion(s) and rate(s) was also adopted in (Muresan and Effros, 2008; Chou *et al.*, 1989; Fleming *et al.*, 2004).

Based on relations (4.4)-(4.6) we obtain that

$$\mathcal{F}(\mathbf{r},\bar{\mathbf{s}}) = \sum_{i=1}^{M_1}\left(\rho\sum_{x\in C_i}(x-\hat{x}(C_i))^2 p_X(x) - (\lambda_1-\lambda_2)P(C_i)\log_2 P(C_i) + \right.$$
$$\left.\sum_{j=1}^{M_{2,i}}\left((1-\rho)\sum_{y\in C_{i,j}}(y-\hat{y}(C_{i,j}|C_i))^2\sum_{x\in C_i}p_{XY}(x,y) - \lambda_2 P(C_i,C_{i,j})\log_2 P(C_i,C_{i,j})\right)\right).$$

In order to simplify the expression of the cost we introduce a few more notations. For each set $C \subseteq \mathcal{X}$ denote

$$d_X(C) \triangleq \rho\sum_{x\in C}(x-\hat{x}(C))^2 p_X(x), \quad h_X(C) \triangleq -(\lambda_1-\lambda_2)P(C)\log_2 P(C).$$

For each $C \subseteq \mathcal{X}$ and $C' \subseteq \mathcal{Y}$ denote

$$d_Y(C'|C) \triangleq (1-\rho)\sum_{y\in C'}(y-\hat{y}(C'|C))^2\sum_{x\in C}p_{XY}(x,y),$$
$$h_Y(C'|C) \triangleq -\lambda_2 P(C,C')\log_2 P(C,C').$$

Using the above notations the cost function in (4.6) becomes

$$\mathcal{F}(\mathbf{r},\bar{\mathbf{s}}) = \sum_{i=1}^{M_1}\left(d_X(C_i) + h_X(C_i) + \underbrace{\sum_{j=1}^{M_{2,i}}\left(d_Y(C_{i,j}|C_i) + h_Y(C_{i,j}|C_i)\right)}_{\tau(C_i,\mathbf{s}_i)}\right).$$

By examining the cost $\mathcal{F}(\mathbf{r},\bar{\mathbf{s}})$ we notice that for each i the contribution of the partition $\mathbf{s}_i$ to the cost function depends on cell C_i, but does not depend on any other cell of the quantizer for X. Therefore, we will denote it by $\tau(C_i,\mathbf{s}_i)$. We conclude that when the partition $\mathbf{r}$ is fixed the optimization of the partition $\mathbf{s}_i$ can be performed separately for each i. In other words, the following holds

$$\min_{M_1,\mathbf{r}\in\mathcal{T}_X(M_1),\bar{\mathbf{s}}\in\mathcal{T}_Y^{M_1}} \mathcal{F}(\mathbf{r},\bar{\mathbf{s}}) = \min_{M_1,\mathbf{r}\in\mathcal{T}_X(M_1)} \sum_{i=1}^{M_1}\left(d_X(C_i) + h_X(C_i) + \min_{M_{2,i},\mathbf{s}_i\in\mathcal{T}_Y(M_{2,i})}\tau(C_i,\mathbf{s}_i)\right).$$

Further, for each $(u,v)\in E_{K_1}$, denote by $\omega(C_X(u,v])$ the minimum value of $\tau(C_i,\mathbf{s}_i)$ over all partitions $\mathbf{s}_i$ when $C_i = C_X(u,v]$, in other words

$$\omega(C_X(u,v]) \triangleq \min_{M_2,\mathbf{s}\in\mathcal{T}_Y(M_2)} \tau(C_X(u,v],\mathbf{s}). \tag{4.7}$$

With the above notation, problem (4.6) becomes equivalent to

$$\min_{M_1,\mathbf{r}\in\mathcal{T}(M_1)} \hat{\mathcal{F}}(\mathbf{r}) \triangleq \sum_{i=1}^{M_1}\left(d_X(C_i) + h_X(C_i) + \omega(C_i)\right). \tag{4.8}$$

We will show that the above problem is equivalent to an MWP problem. Indeed, consider the WDAG $G_X(w)$, where, for each $(u,v)\in E_{K_1}$, $w(u,v)$ is defined by

$$w(u,v) \triangleq d_X(C_X(u,v]) + h_X(C_X(u,v]) + \omega(C_X(u,v]). \tag{4.9}$$

Then any partition $\mathbf{r} \in \mathcal{T}_X(M_1)$ is in a one-to-one correspondence with an M_1-edge path in $G_X(w)$, from the source to the final node. Additionally, the weight of the path equals the cost $\hat{\mathcal{F}}(\mathbf{r})$. This implies that problem (4.8) is equivalent to the MWP problem in $G_X(w)$.

In order to solve the MWP problem in $G_X(w)$ we need to be able to evaluate each edge weight. Therefore, we need to solve first problem (4.7) for each edge (u, v). It turns out that problem (4.7) is also equivalent to an MWP problem in some other WDAG. Indeed, consider the WDAG $G_Y(w_{u,v})$, where for each edge $(m, n) \in E_{K_2}$, the weight $w_{u,v}(m, n)$ is defined as follows

$$w_{u,v}(m,n) \triangleq d_Y(C_Y(m,n]|C_X(u,v]) + h_Y(C_Y(m,n]|C_X(u,v]). \tag{4.10}$$

Then any partition $\mathbf{s} \in \mathcal{T}_Y(M_2)$ is in a one-to-one correspondence with an M_2-edge path from the source to the final node in WDAG $G_Y(w_{u,v})$. The weight of the path equals the cost function in (4.7), thus problem (4.7) is equivalent to the MWP path problem in $G_Y(w_{u,v})$.

Notice that solving the MWP problem in some WDAG requires $O(|V| + |E|)$ operations, if the weight of each edge can be evaluated in constant time, where V denotes the vertex set and E denotes the edge set. In order to enable the evaluation in constant time of each edge weight, we include a preprocessing step which computes and stores the following cumulative values

$$\phi_{k,X}(u) \triangleq \sum_{i=1}^{u} x^k p_X(x_i), \qquad \phi_{k,XY}(u,m) \triangleq \sum_{j=1}^{m}\sum_{i=1}^{u} y^k p_{XY}(x_i, y_j),$$

for $k = 0, 1, 2$, $0 \leq u \leq K_1$ and $0 \leq m \leq K_2$. All the above values can be computed in $O(K_1K_2)$ time, while the amount of memory needed store all of them is also $O(K_1K_2)$.

Then $P(C_X(u,v], C_Y(m,n])$ can be computed in constant time as follows

$$P(C_X(u,v], C_Y(m,n]) = \phi_{0,XY}(v,n) - \phi_{0,XY}(v,m) - \phi_{0,XY}(u,n) + \phi_{0,XY}(u,m).$$

Similarly, the quantity $\sum_{j=m+1}^{n} \sum_{i=u+1}^{v} y_j p_{XY}(x_i, y_j)$ can be evaluated in constant time using $\phi_{1,XY}(\cdot,\cdot)$, leading further to the evaluation of $\hat{y}(C_Y(m,n]|C_X(u,v])$ in $O(1)$ time as well. Next notice that

$$\sum_{j=m+1}^{n} (y_j - \hat{y}(C_Y(m,n]|C_X(u,v]))^2 \sum_{i=u+1}^{v} p_{XY}(x_i, y_j) =$$
$$\sum_{j=m+1}^{n} \sum_{i=u+1}^{v} y_j^2 p_{XY}(x_i, y_j) - \hat{y}(C_Y(m,n]|C_X(u,v])^2 P(C_X(u,v], C_Y(m,n]),$$

where $\sum_{j=m+1}^{n} \sum_{i=u+1}^{v} y_j^2 p_{XY}(x_i, y_j)$ can also be computed in $O(1)$ time based on $\phi_{2,XY}(\cdot,\cdot)$.

Let us summarize now the solution algorithm to problem (4.6). After performing the preprocessing step the algorithm proceeds in two stages as follows.

1) For each pair $(u,v) \in E_{K_1}$, solve the MWP problem in $G_Y(w_{u,v})$, where $w_{u,v}$ is given in (4.10). This takes $O(K_2^2)$ operations for each pair (u,v). Doing so for all $(u,v) \in E_{K_1}$ amounts to $O(K_1^2 K_2^2)$ operations.

2) Solve the MWP problem in $G_X(w)$, where w is given in (4.9). This can be done in $O(K_1^2)$ time.

In conclusion, the overall time complexity of the proposed algorithm is $O(K_1^2 K_2^2)$.

4.3 Optimal FR-SSQ Design Algorithm

In this section, we formulate the optimal FR-SSQ design problem and present its solution.

The rates in the FR case are

$$\begin{aligned} R_1(\mathbf{r}) &= \log_2 M_1, \\ R_2(\mathbf{r}, \bar{\mathbf{s}}) &= \sum_{i=1}^{M_1} P(C_i) \log_2 M_{2,i}. \end{aligned} \tag{4.11}$$

It is easy to impose a constraint $R_1(\mathbf{r}) \leq R_1$ on the rate of encoder 1 by fixing the number of cells in Q_1 to be

$$M_1 = \lfloor 2^{R_1} \rfloor. \tag{4.12}$$

The problem of optimal FR-SSQ design is formulated as

$$\min_{\mathbf{r} \in \mathcal{T}_X(M_1), \bar{\mathbf{s}} \in \mathcal{T}_Y^{M_1}} \mathcal{F}'(\mathbf{r}, \bar{\mathbf{s}}) \triangleq \rho D_1(\mathbf{r}) + (1-\rho) D_2(\mathbf{r}, \bar{\mathbf{s}}) + \lambda_2 R_2(\mathbf{r}, \bar{\mathbf{s}}), \tag{4.13}$$

for some fixed $\rho, 0 < \rho < 1$, and $\lambda_2 > 0$. Let $\mathcal{RD}_{FR}(R_1)$ denote the set of quadruples $(R_1(\mathbf{r}), R_2(\mathbf{r}, \bar{\mathbf{s}}), D_1(\mathbf{r}), D_2(\mathbf{r}, \bar{\mathbf{s}}))$ satisfying (4.12). Then any point on the lower boundary of the convex hull of $\mathcal{RD}_{FR}(R_1)$ can be obtained by solving problem (4.13) for some choice of ρ and λ_2 as above.

Using the notations introduced in the previous section, the cost in (4.13) becomes

$$\mathcal{F}'(\mathbf{r}, \bar{\mathbf{s}}) = \sum_{i=1}^{M_1} \Bigg(d_X(C_i) + \underbrace{\lambda_2 P(C_i) \log_2 M_{2,i} + \sum_{j=1}^{M_{2,i}} d_Y(C_{i,j} | C_i)}_{\tau'(C_i, \mathbf{s}_i)} \Bigg).$$

Similarly to the EC case, if cell C_i is fixed the partition $\mathbf{s}_i$ can be optimized by minimizing the cost $\tau'(C_i, \mathbf{s}_i)$. Therefore, for each $(u, v) \in E_{K_1}$ let us denote by

$\omega'(C_X(u,v])$ the minimum value of $\tau'(C_i, \mathbf{s}_i)$ over all $\mathbf{s}_i$ when $C_i = C_X(u,v]$, i.e.,

$$\omega'(C_X(u,v]) \triangleq \min_{M_2, \mathbf{s}\in\mathcal{T}_Y(M_2)} \tau'(C_X(u,v], \mathbf{s}). \tag{4.14}$$

Then problem (4.13) becomes equivalent to

$$\min_{\mathbf{r}\in\mathcal{T}_X(M_1)} \hat{\mathcal{F}}'(\mathbf{r}) \triangleq \sum_{i=1}^{M_1} \left(d_X(C_i) + \omega'(C_i)\right). \tag{4.15}$$

Consider the WDAG $G_X(w')$, where for each $(u,v) \in E_{K_1}$ the weight $w'(u,v)$ is defined as

$$w'(u,v) = d_X(C_X(u,v]) + \omega'(C_X(u,v]). \tag{4.16}$$

Then any ascending M_1-sequence $\mathbf{r}$ can be identified with an M_1-edge path in $G_X(w')$ from the source to the final node, whose weight equals the cost $\hat{\mathcal{F}}'(\mathbf{r})$. Since the correspondence is one-to-one, it follows that problem (4.15) is equivalent to the M_1-edge MWP problem in $G_X(w')$.

In order to solve the aforementioned problem we need to determine first the value of $\omega'(C_X(u,v])$ by solving the minimization in (4.14), for each $(u,v) \in E_{K_1}$. Note that, unlike its counterpart (4.7) in the EC case, problem (4.14) can no longer be cast as an MWP problem. In order to solve it notice that the following holds

$$\omega'(C_X(u,v]) = \min_{M_2} \left(\lambda_2 P(C_X(u,v]) \log_2 M_2 + \underbrace{\min_{\mathbf{s}\in\mathcal{T}_Y(M_2)} \sum_{j=1}^{M_2} d_Y(C'_j | C_X(u,v])}_{\hat{W}_{u,v}(M_2)} \right). \tag{4.17}$$

We conclude that the above problem can be solved in two stages.

A) Solve first the inner minimization over ascending M_2-sequences $\mathbf{s}$, for each positive integer M_2.

B) Solve the outer minimization over positive integers M_2.

For each $M_2 > 0$ the inner minimization is equivalent to the M_2-edge MWP problem in the WDAG $G_Y(w'_{u,v})$, where for each $(m,n) \in E_{K_2}$ the weight $w'_{u,v}(m,n)$ is defined as follows

$$w'_{u,v}(m,n) \triangleq d_Y(C_Y(m,n]|C_X(u,v]).$$

Thus, the quantity $\hat{W}_{u,v}(M_2)$ defined in (4.17) equals the weight of the M_2-edge MWP in $G_Y(w'_{u,v})$. As pointed out above, solving (4.17) can be done by determining $\hat{W}_{u,v}(M_2)$ for each M_2 and then performing a linear search over M_2.

The computation of $\hat{W}_{u,v}(M_2)$ can be accomplished using dynamic programming (DP). The DP algorithm finds the k-edge MWP path from node 0 to node n for each pair (k,n) with $1 \leq k \leq M_2$ and $1 \leq n \leq K_2$. Let $W_{u,v}(k,n)$ denote the weight of the k-edge MWP path from node 0 to node n. Then the following recurrence relation holds for all $2 \leq k \leq M_2$ and $2 \leq n \leq K_2$

$$W_{u,v}(k,n) = \min_{1 \leq m < n} \left(W_{u,v}(k-1,m) + w'_{u,v}(m,n) \right). \tag{4.18}$$

Clearly, $W_{u,v}(1,m) = w'_{u,v}(0,m)$ for all $m \in I_{K_2} \setminus \{0\}$. The DP process solves (4.18) for all pairs (k,n), $1 \leq k \leq M_2$, $1 \leq n \leq K_2$, in lexicographical order. The value $\hat{W}_{u,v}(M_2)$ sought of equals $W_{u,v}(M_2,K_2)$. The total amount of operations reaches $O(M_2 K_2^2)$.

Note that the above procedure to solve the M_2-edge MWP problem, also solves the k-edge MWP problem for all smaller path lengths k, for $1 \leq k < M_2$. Since the maximum possible value of M_2 is K_2, it follows that solving the M_2-edge MWP

problem for all $1 \leq M_2 \leq K_2$ can be done in $O(K_2^3)$ time. Since the additional linear search over M_2 in (4.17) takes only $O(K_2)$ time, it follows that problem (4.17) can be solved in $O(K_2^3)$ time.

Next we will show that the edge weights in the WDAG $G_Y(w'_{u,v})$ satisfy the so-called Monge property, fact which allows for a speed-up of the DP algorithm.

Lemma. The edge weights in the WDAG $G_Y(w'_{u,v})$ satisfy the Monge property, i.e., the following holds

$$\begin{aligned} &w'_{u,v}(m,n) + w'_{u,v}(m',n') \leq w'_{u,v}(m,n') + w'_{u,v}(m',n), \\ &\text{for all } 0 \leq m < m' < n < n' \leq K_2. \end{aligned} \tag{4.19}$$

Proof:

Let $C = C_X(u,v]$, $p_C(y) \triangleq \frac{\sum_{x \in C} p_{XY}(x,y)}{P(C)}$ and

$$\eta(m,n) \triangleq \sum_{j=m+1}^{n} (y_j - \hat{y}(C_Y(m,n]|C))^2 p_C(y_j).$$

Then we have

$$w'_{u,v}(m,n) = (1-\rho) P(C_X(u,v]) \eta(m,n). \tag{4.20}$$

Note that $p_C(y)$ is a pmf and $\hat{y}(C_Y(m,n]|C) = \frac{\sum_{j=m+1}^{n} y_j p_C(y_j)}{\sum_{j=m+1}^{n} p_C(y_j)}$. Then according to (Wu, 1991; Wu and Zhang, 1993) the function $\eta(m,n)$ satisfies the Monge property, i.e., the following holds

$$\begin{aligned} &\eta(m,n) + \eta(m',n') \leq \eta(m,n') + \eta(m',n), \\ &\text{for all } 0 \leq m < m' < n < n' \leq K_2. \end{aligned}$$

The above property in conjunction with (4.20) implies (4.19), thus completing the proof.

Since the weights $w'_{u,v}(m,n)$ of the WDAG $G_Y(w'_{u,v})$ satisfy the Monge property, the DP algorithm used to solve the problem at stage A can be sped up by a factor of K_2 (Wu, 1991; Wu and Zhang, 1993). Specifically, this is done by applying the so-called SMAWK algorithm introduced in (Aggarwal *et al.*, 1987) to compute all values $W_{u,v}(k,n)$ for all n and fixed k, in $O(K_2)$ operations. This implies that problem (4.17) can be solved in $O(K_2^2)$ time. It follows that computing $\omega'(C_X(u,v])$ for all pairs $(u,v) \in E_{K_1}$ takes $O(K_1^2 K_2^2)$ operations.

Let us summarize now the proposed solution to the optimal FR-SSQ design problem (4.13). We start with a preprocessing step as in the EC case. After that the algorithm proceeds as follows.

Step 1) For each pair $(u,v) \in E_{K_1}$, solve problem (4.17) in the following two stages.

A) Solve the M_2-edge MWP problem in $G_Y(w'_{u,v})$, for all $1 \le M_2 \le K_2$. To this end, for each $1 \le k \le K_2$ use SMAWK to compute $W_{u,v}(n)$ for all $1 \le n \le K_2$.

B) Compute $\omega'(C_X(u,v]) = \min_{M_2} \left(\lambda_2 P(C_X(u,v]) \log_2 M_2 + \hat{W}_{u,v}(M_2) \right)$.

Step 2) Solve the M_1-edge MWP problem in the WDAG $G_X(w')$.

Recall that the preprocessing step needs $O(K_1 K_2)$ time. Further, Step 1 requires $O(K_1^2 K_2^2)$ operations. Step 2 can be accomplished in $O(M_1 K_1^2)$ running time. In conclusion, the overall running time to solve problem (4.13) is $O(K_1^2 K_2^2)$ assuming that $M_1 = O(K_2^2)$.

4.4 Application to Continuous Sources

In this section we assume that the sources X and Y are continuous and apply the proposed algorithms to discretized versions of X and Y. We show that the EC-SSQ, respectively FR-SSQ, obtained in this way approaches in performance the optimal EC-SSQ, respectively FR-SSQ, with convex cells for the original sources as the discretization increases in accuracy.

First we need to introduce some notations. For any pair of real-valued RVs (X, Y) with joint pdf f_{XY}, for each positive real value B and positive integer K, we define the pair continuous RVs (X_B, Y_B) and the pair of discrete RVs $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$ as follows. (X_B, Y_B) is the truncation of (X, Y) to the set $[-B, B] \times [-B, B]$, i.e., its pdf is $f_{X_B Y_B}(x, y) \triangleq \frac{f_{XY}(x,y)}{\int_{-B}^{B}\int_{-B}^{B} f_{XY}(x,y)\ dx\ dy}$ when $(x, y) \in [-B, B] \times [-B, B]$ and 0 otherwise. The marginal pdfs of X_B and Y_B are denoted by f_{X_B} and f_{Y_B}, respectively. Further, $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$ is the quantized version of (X_B, Y_B) using a product scalar quantizer. More specifically, each scalar quantizer has K cells of equal size, and the centroid of each cell as the reconstruction value. Thus, the thresholds of each scalar quantizer are $t_0^{(B)}, \ldots, t_K^{(B)}$, where $t_k^{(B)} \triangleq -B + \frac{2kB}{K}$, $0 \le k \le K$. Let $\mathcal{U}_{B,K}$ denote the set of these thresholds. The alphabet of $\tilde{X}_{B,K}$ is $\tilde{\mathcal{X}}_{B,K} = \{x_k^{(B)} \triangleq \frac{\int_{t_{k-1}^{(B)}}^{t_k^{(B)}} x f_{X_B}(x)\ dx}{\int_{t_{k-1}^{(B)}}^{t_k^{(B)}} f_{X_B}(x)\ dx} | 1 \le k \le K\}$. The alphabet of $\tilde{Y}_{B,K}$ is $\tilde{\mathcal{Y}}_{B,K} = \{y_k^{(B)} \triangleq \frac{\int_{t_{k-1}^{(B)}}^{t_k^{(B)}} y f_{Y_B}(y)\ dy}{\int_{t_{k-1}^{(B)}}^{t_k^{(B)}} f_{Y_B}(y)\ dy} | 1 \le k \le K\}$. The joint pmf of $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$ is $P_{\tilde{X}_{B,K}\tilde{Y}_{B,K}}(x_k^{(B)}, y_l^{(B)}) \triangleq \int_{t_{k-1}^{(B)}}^{t_k^{(B)}} \int_{t_{l-1}^{(B)}}^{t_l^{(B)}} f_{X_B Y_B}(x, y)\ dy\ dx$, $1 \le k, l \le K$.

An SSQ for a continuous source is specified by the encoding functions f_1, f_2 and the decoding functions g_1, g_2, as in (4.1) where $\mathcal{I}_1 = \{1, 2, \ldots, M_1\}$ or $\mathcal{I}_1 = \mathbb{Z}$ and $\mathcal{I}_2 = \{1, 2, \ldots, M_2\}$ or $\mathcal{I}_2 = \mathbb{Z}$. Note that we also consider the possibility that $\mathcal{I}_1 = \mathbb{Z}$ and $\mathcal{I}_2 = \mathbb{Z}$ in the EC case. We consider SSQs with convex cells, thus the cells of each

partition are intervals, open at the left end and closed at the right end (except when the right end equals infinity). They are labeled in increasing order from left to right. Additionally, we only consider partitions where the number of cells is finite in any bounded interval[2]. For simplicity, let us denote $\mathbf{Q} = (f_1, f_2, g_1, g_2)$. When applying the SSQ $\mathbf{Q}$ to a pair of RVs (X', Y'), we denote by $D_1(\mathbf{Q}, X')$ and $D_2(\mathbf{Q}, X', Y')$ the distortions at the first and second decoder, respectively, i.e.,

$$D_1(\mathbf{Q}, X') \triangleq \mathbb{E}[(X' - \hat{X}')^2], \quad D_2(\mathbf{Q}, X', Y') \triangleq \mathbb{E}[(Y' - \hat{Y}')^2],$$

where $\hat{X}' = g_1(f_1(X'))$ and $\hat{Y}' = g_2(f_1(X'), f_2(f_1(X'), Y'))$. The rates of the two encoders in the EC case will be denoted by $R_{EC,1}(\mathbf{Q}, X')$ and $R_{EC,2}(\mathbf{Q}, X', Y')$, respectively. Thus,

$$R_{EC,1}(\mathbf{Q}, X') \triangleq -\mathbb{E}[\log_2 P(f_1(X'))], \quad R_{EC,2}(\mathbf{Q}, X', Y') \triangleq -\mathbb{E}[\log_2 P(f_2(f_1(X'), Y')|f_1(X'))],$$

where, for a discrete RV $\tilde{Z}$, $P(\tilde{Z})$ denotes its pmf, i.e., $P(\tilde{Z}) = p_{\tilde{Z}}(\tilde{Z})$. The rates in the FR case are $R_{FR,1}(\mathbf{Q}, X') \triangleq \log_2 M_1$ and $R_{FR,2}(\mathbf{Q}, X', Y') \triangleq -\mathbb{E}[\log_2 M_{2,I}]$. Note that in the FR case we necessarily have $\mathcal{I}_1$ and $\mathcal{I}_2$ finite. We denote by $\mathbf{\Omega}_{EC}$ and by $\mathbf{\Omega}_{FR}$ the class of EC-SSQs and of FR-SSQs defined as above, respectively. Finally, consider fixed $0 < \rho < 1$, $\lambda_1 > 0$ and $\lambda_2 > 0$ and denote

$$\mathcal{F}_{EC}(\mathbf{Q}, X', Y') \triangleq \rho D_1(\mathbf{Q}, X') + (1-\rho) D_2(\mathbf{Q}, X', Y') + \lambda_1 R_{EC,1}(\mathbf{Q}, X') + \lambda_2 R_{EC,2}(\mathbf{Q}, X', Y'),$$

$$\mathcal{F}_{FR}(\mathbf{Q}, X', Y') \triangleq \rho D_1(\mathbf{Q}, X') + (1-\rho) D_2(\mathbf{Q}, X', Y') + \lambda_2 R_{FR,2}(\mathbf{Q}, X', Y'),$$

[2]Note that considering only partitions where the number of cells is finite in any bounded interval does not preclude the optimality of the quantizer for Y, according to (Gyorgy *et al.*, 2003). It might also be possible that the arguments of (Gyorgy *et al.*, 2003) could be extended to prove a similar claim for the quantizer for X. The investigation of this possibility is left for future work.

The proof of the following result is deferred to appendix D.

Theorem 4.1. Let (X, Y) be a pair of jointly distributed real-valued RVs with a continuous joint pdf f_{XY} with finite variance, satisfying $f_{XY}(x, y) > 0$ for any $x, y \in \mathbb{R}$. Let

$$\mathcal{F}^*_{EC} \triangleq \inf_{\mathbf{Q} \in \mathbf{\Omega}_{EC}} \mathcal{F}_{EC}(\mathbf{Q}, X, Y). \tag{4.21}$$

For each positive real value B and positive integer K, let $\hat{\mathbf{Q}}_{B,K}$ denote the optimal EC-SSQ with convex cells for the pair of discrete RVs $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$. Then the following holds

$$\lim_{B\to\infty} \lim_{K\to\infty} \mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K}, \tilde{X}_{B,K}, \tilde{Y}_{B,K}) = \mathcal{F}^*_{EC}. \tag{4.22}$$

Furthermore, for each positive integer M_1, let

$$\mathcal{F}^*_{FR}(M_1) \triangleq \inf_{\mathbf{Q} \in \mathbf{\Omega}_{FR}} \mathcal{F}_{FR}(\mathbf{Q}, X, Y).$$

For each positive real value B and positive integer K, let $\hat{\mathbf{Q}}_{B,K}(M_1)$ denote the optimal FR-SSQ with convex cells for the pair of discrete RVs $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$. Then the following holds

$$\lim_{B\to\infty} \lim_{K\to\infty} \mathcal{F}_{FR}(\hat{\mathbf{Q}}_{B,K}(M_1), \tilde{X}_{B,K}, \tilde{Y}_{B,K}) = \mathcal{F}^*_{FR}(M_1). \tag{4.23}$$

4.5 Experimental Results and Discussion

This section assesses the practical performance of the proposed EC-SSQ and FR-SSQ design algorithms for discretized Gaussian sources. We start with a pair (X, Y)

of correlated Gaussian sources, both with 0 mean and variance 1, with joint pdf

$$f_{XY}(x,y) = \frac{1}{2\pi\sqrt{1-c^2}} \exp\left(-\frac{x^2+y^2-2xyc}{2(1-c^2)}\right),$$

where c is the correlation coefficient. We consider $c = 0.5$ and $c = 0.9$ in this section.

Next we consider the discrete sources $\tilde{X} = \tilde{X}_{B_1,K_1}$ and $\tilde{Y} = \tilde{Y}_{B_2,K_2}$, where $B_1 = 3$, $B_2 = 5$, $K_1 = 100$ and $K_2 = 160$. The proposed EC-SSQ and FR-SSQ design algorithms are applied to the pair of discrete sources $(\tilde{X}, \tilde{Y})$ and the obtained SSQs are extended to SSQs for the continuous sources (X, Y). Then the distortions at the two decoders, denoted by D_1, respectively D_2, and the rates of the two encoders, denoted by R_1, respectively R_2, are evaluated for the extended SSQs applied to (X, Y).

An SSQ for the discrete sources $(\tilde{X}, \tilde{Y})$ is extended to an SSQ for (X, Y) by extending each partition of the alphabet of $\tilde{X}$ and each partition of the alphabet of $\tilde{Y}$ to a partition of $\mathbb{R}$ with the same number of cells as follows. A partition for $\tilde{X}$ specified by the sequence of thresholds $0 = r_0 < r_1 < \ldots < r_{M_1} = K_1$ is extended to the partition of $\mathbb{R}$ with thresholds $(-\infty, t^{(B_1)}_{r_1}, \ldots, t^{(B_1)}_{r_{M_1-1}}, \infty)$. Likewise, a partition for $\tilde{Y}$ specified by the sequence of thresholds $0 = s_0 < s_1 < \ldots < s_{M_2} = K_2$ is extended to the partition of $\mathbb{R}$ with thresholds $(-\infty, t^{(B_2)}_{s_1}, \ldots, t^{(B_2)}_{s_{M_1-1}}, \infty)$.

We first consider the case of EC-SSQ. We ran the proposed algorithm for optimal EC-SSQ design for four values of ρ, namely, $\rho = 0.1, 0.5, 0.9, 0.95$ and a large set of values of λ_1 and λ_2 with $\lambda_1 \in [0.01, 1.50]$ and $\lambda_2 \in [0.01, 1.0]$.

Figures 4.1 and 4.2 illustrate the performance comparison against the theoretical rate-distortion bounds. Figures 4.1a and 4.2a plot the distortion pairs (D_1, D_2) obtained in our experiments for $c = 0.9$ and $c = 0.5$, respectively. Each figure also shows the boundary of the theoretical region of nontrivial distortion pairs, which is characterized by $0 \leq D_1 \leq 1$ and $D_2 \leq 1 - c^2(1 - D_1)$. These figures show that,

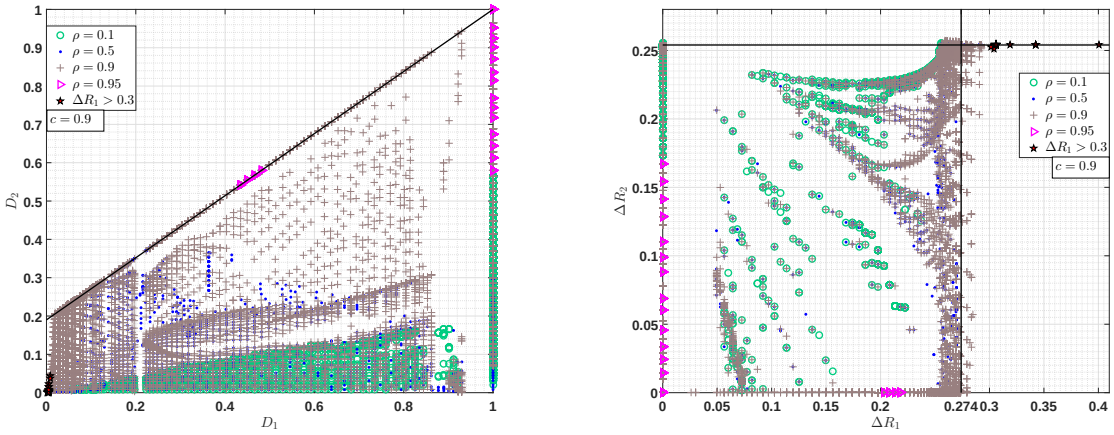
by varying the parameters ρ, λ_1 and λ_2 the proposed design is able to achieve a dense set of distortion pairs covering fairly well the theoretical distortion region. For each distortion pair (D_1, D_2) achieved by our scheme we compute the rate-gap pair $(\Delta R_1, \Delta R_2)$ relative to the theoretical lower bound, namely $\Delta R_i = R_i - R_i^*$, $i = 1, 2$, where (R_1^*, R_2^*) denotes the pair of information theoretical lower bounds on the rates at the two encoders for the distortion pair (D_1, D_2). According to (Viswanathan and Berger, 2000), we have

$$R_1^* = \frac{1}{2}\log_2 \frac{1}{D_1},$$
$$R_2^* = \frac{1}{2}\log_2 \frac{1 - c^2(1 - D_1)}{D_2}.$$

The rate-gap pairs are plotted in Figures 4.1b and 4.2b for $c = 0.9$ and $c = 0.5$, respectively. Recall that a rate gap of 0.2546 bits/sample is expected at high resolution, as explained in Chapter 3.

As it can be seen from Figures 4.1b and 4.2b in most of the cases the rate-gap at encoder 2 is within 0.254 bits/sample, while the gap at encoder 1 is within 0.274 bits/sample, which is very close to the gap due to the low dimensionality of the EC-SSQ. This fact demonstrates the effectiveness of the proposed EC-SSQ design algorithm as an approximation of the optimal EC-SSQ for continuous sources.

We also mention that the largest value of ΔR_2 is only slightly higher than the benchmark value of 0.254, namely it is 0.257 bits/sample for $c = 0.9$, respectively 0.262 bits/sample for $c = 0.5$. On the other hand, there are several cases for which the rate-gap at encoder 2 ranges between 0.3 and 0.4. The corresponding rate-gap pairs and distortion pairs are marked using star-shaped markers in Figures 4.1 and 4.2. We observe that these cases with excess rate loss are obtained when D_1 is very small (thus, R_1 is very high), while $D_2 \lesssim 0.1$. One possible reason for this additional

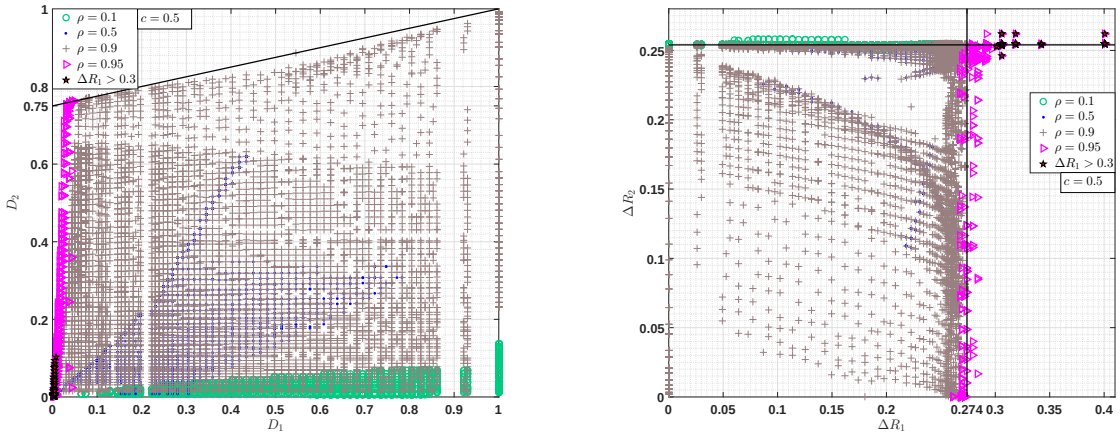


(a) Distortion pairs against the theoretical bound. (b) Gap to the theoretical minimum rate.

Figure 4.1: Comparison between the achievable rate-distortion performance and the theoretical bound for $c = 0.9$.

rate loss could be the coarseness of discretization for the source X. Another possible reason could be the additional tension in the optimization of encoder 1 generated by the competing requirements at the two decoders. Namely, there is tension between ensuring a good reconstruction of the source X as well as facilitating an efficient encoder for the source Y.

It is also interesting to investigate the impact that the refinement of the discretization has on the EC-SSQ performance. Table 4.1 compares the rate-distortion performance for four pairs K_1, K_2 representing a gradual increase in the discretization accuracy. The EC-SSQ design algorithm is applied in all four cases to the same parameters, namely $c = 0.9$, $\lambda_1 = 0.22$, $\lambda_2 = 0.15$ and $\rho = 0.5$. The pair $K_1 = 100, K_2 = 160$ represents the coarsest discretization. The discretization is refined gradually by multiplying the initial values of K_1 and K_2 by two, five and ten, respectively. It can be noted that the rate gaps generally decrease, as expected, but the decrease is very small. In particular, the relative decrease of ΔR_1 from the initial to the final value is of 0.5%, while for ΔR_2 the relative decrease is of 0.28%.



(a) Distortion pairs against the theoretical bound. (b) Gap to the theoretical minimum rate.

Figure 4.2: Comparison between the achievable rate-distortion performance and the theoretical bound for $c = 0.5$.

(K_1, K_2)	$(100, 160)$	$(200, 320)$	$(500, 800)$	$(1000, 1600)$
R_1	1.3173	1.3030	1.3059	1.3030
D_1	0.2307	0.2349	0.2340	0.2349
R_2	1.0430	1.0461	1.0442	1.0452
D_2	0.1196	0.1201	0.1201	0.1202
ΔR_1	0.2593	0.2580	0.2582	0.2580
ΔR_2	0.2150	0.2145	0.2144	0.2144

Table 4.1: Rate-distortion performance comparison of the proposed EC-SSQ for various K_1 and K_2.

It is instructive to analyze the structure of the encoder partitions generated by the proposed approach. Note that in the sequel the distortion is represented in dB, i.e., as $10 \log_{10} D$. Figure 4.3 illustrates the optimized encoder partitions of the proposed EC-SSQ with $R_1 = 1.3173$ and $R_2 = 1.0430$, when $c = 0.9$ and $\rho = 0.5$. In this example, the source X is quantized to $M_1 = 3$ cells with sequence of thresholds $(-\infty, -0.9, 0.9, \infty)$. For each $i = 1, 2, 3$, all quantizers of source Y have $M_{2,i} = 4$ cells. The partitions corresponding to the quantizers for Y, for $i = 1, 2, 3$, are defined by the sequences of thresholds $(-\infty, -3.1250, -1.75, -0.1875, \infty)$,

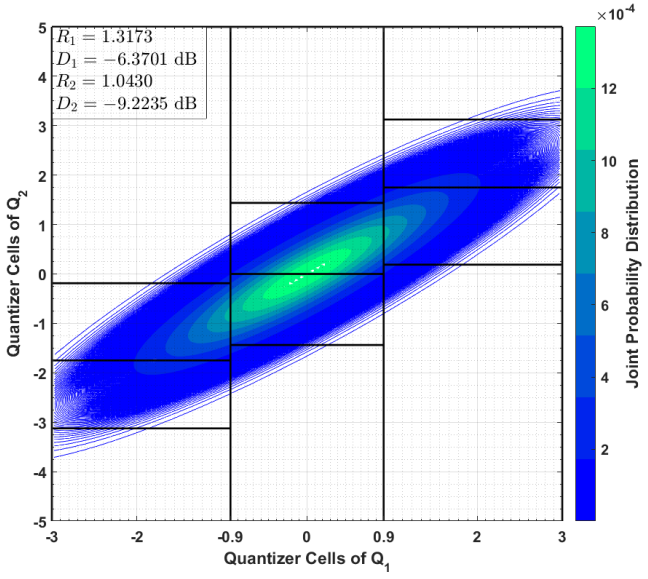


Figure 4.3: Example of optimized encoder partitions of the proposed EC-SSQ, when $c = 0.9$ and $\rho = 0.5$.

$(-\infty, -1.4375, 0.0, 1.4375, \infty)$ and $(-\infty, 0.1875, 1.75, 3.125, \infty)$, respectively. In addition, the contour of the joint pdf f_{XY} is also plotted in Figure 4.3, where the probability decreases as the color changes from green to blue. It is worth pointing out that the output of the quantizer of Y is more densely spaced where the joint probability takes on large values, as expected.

Figure 4.4 plots the distortion D_2 of the proposed EC-SSQ, versus the rate R_2, when the pair (R_1, D_1) is fixed, for three cases of (R_1, D_1) with $c = 0.9$ and $\rho = 0.5$. As expected, for fixed pair (R_1, D_1), the distortion at the second decoder decreases steadily as the rate at encoder 2 increases. On the other hand, when the rate R_2 is kept fixed, the performance at decoder 2 also improves consistently with the increase of the rate at encoder 1. In particular, when R_1 increases from 1.1983 to 1.6095, the performance at decoder 2 jumps up by about 0.9 dB. The further increase of R_1 to 1.9367 leads to another gain of about 0.5 dB at decoder 2. This is expected since increasing R_1, intuitively, corresponds to refining the information about the source

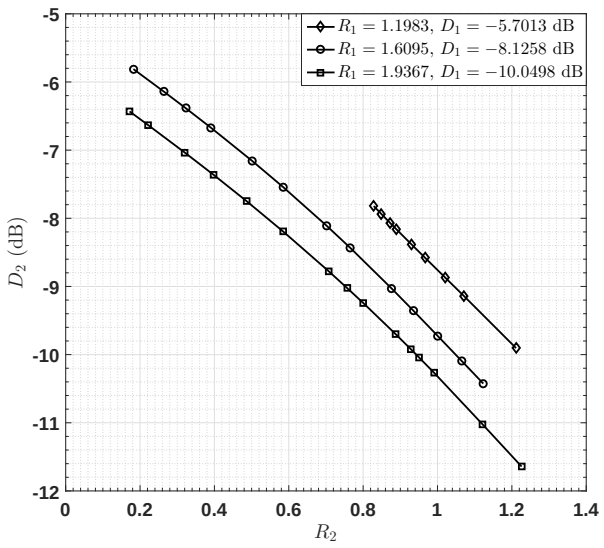


Figure 4.4: Performance of proposed EC-SSQ at decoder 2, for three pairs (R_1, D_1) when $c = 0.9$ and $\rho = 0.5$.

X. Since X and Y are correlated, the refinement of the information about X leads to more information about Y. Thus, the rate at encoder 2 is used to further refine the information which is already available about Y through the reconstruction of X.

Next we assess the performance of the proposed FR-SSQ design algorithm in comparison with the level-constrained practical SSQ scheme developed in (Balasubramanian *et al.*, 1995) based on the asymptotic quantization theory. The authors of (Balasubramanian *et al.*, 1995) use the following quantizer density functions for Q_1, respectively $Q_{2,i}$,

$$\lambda(x) = \frac{f_X(x)^{1/3}}{\int f_X(x)^{1/3} dx},$$
$$\lambda(y|C_i) = \frac{f(y|C_i)^{1/3}}{\int f(y|C_i)^{1/3} dy},$$

to derive the asymptotical expressions of the distortion, as the rates approach infinity. Further, based on the asymptotical analysis, they propose a practical scheme

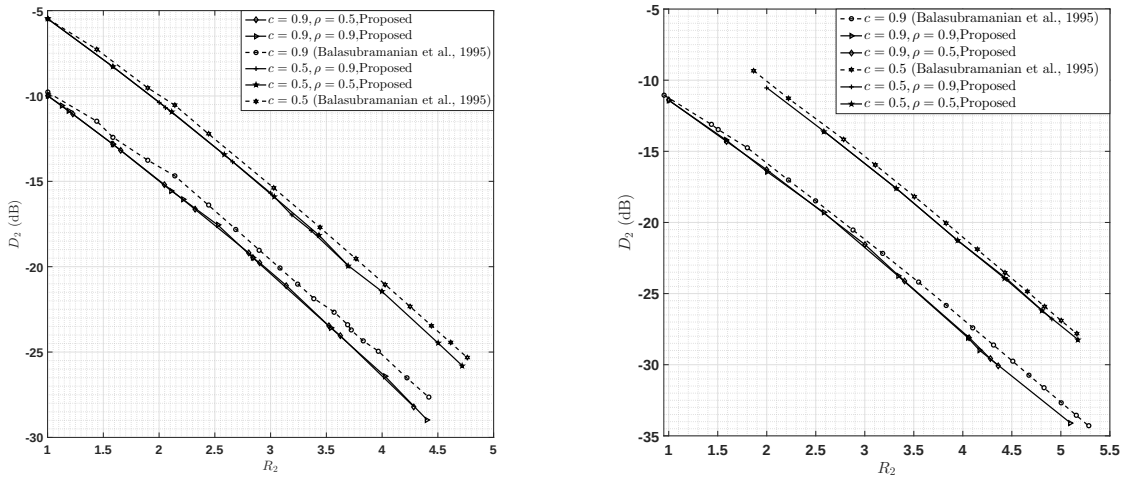
operating at finite rates. Note that the design of (Balasubramanian *et al.*, 1995) is performed under the constraint that $\sum_{i=1}^{M_1} M_{2,i} = N$, for some target value N. The practical construction of (Balasubramanian *et al.*, 1995) proceeds as follows. First, the encoding function f_1 partitions the real line into M_1 cells such that the area under the function $\lambda(x)$ within each cell equals $1/M_1$, using the marginal pdf $f_X(x)$. Subsequently, the values of $M_{2,i}$ are computed using

$$M_{2,i} = \left\lfloor N \frac{[||f(y|C_i)||_{1/3} P(C_i)]^{1/3}}{\sum_{i=1}^{M_1} [||f(y|C_i)||_{1/3} P(C_i)]^{1/3}} \right\rceil,$$

where $||f(x)||_m = [\int f(x)^m dx]^{1/m}$, while$\lfloor \cdot \rceil$ denotes rounding to the nearest integer. Subsequently, for each cell $i, 1 \leq i \leq M_1$, the encoding function $f_2(i, \cdot)$ partitions the real line into $M_{2,i}$ cells such that the area under $\lambda(y|C_i)$ within each cell equals $1/M_{2,i}$, using the conditional pdf $f(y|C_i)$. Finally, the reconstruction values are taken as the centroid of each quantization cell. The distortion and the average rate of quantizer Q_2 are evaluated using (4.4) and (4.11), respectively. To implement the practical FR-SSQ based on the above asymptotic analysis, the same discretization procedure as for the proposed algorithm is utilized with $K_1 = 3000$ and $K_2 = 5000$.

We ran the proposed algorithm for optimal FR-SSQ design for two values of M_1, namely 4 and 16, for $\rho = 0.5, 0.9$ and a set of values of λ_2 satisfying $\lambda_2 \in [0.00001, 0.05]$.

Figures 4.5a and 4.5b plot the distortion D_2 versus the average rate R_2, for the proposed FR-SSQ in comparison with the scheme of (Balasubramanian *et al.*, 1995), for $M_1 = 4$ and $M_1 = 16$, respectively. The plots for both correlation coefficients $c = 0.5$ and $c = 0.9$ and $\rho = 0.5, 0.9$ are included. It can be observed from both figures that the performance when $\rho = 0.5$ and $\rho = 0.9$ is almost identical. It can also be seen that our design always outperforms the scheme of (Balasubramanian *et al.*, 1995). To make the comparison easier, we show in Tables 4.2 and 4.3 the performance



(a) $M_1 = 4$. (b) $M_1 = 16$.

Figure 4.5: Performance comparison of the proposed FR-SSQ against the level-constrained SSQ of (Balasubramanian *et al.*, 1995).

improvement (in dB) over the scheme of (Balasubramanian *et al.*, 1995) at decoder 2 for various values of R_2, when $M_1 = 4$ and 16, respectively. Note that when $R_2 \approx 1.0$, the quantizers of Y for all the schemes have $M_{2,i} \leq 2$ cells. This explains why the improvement is small at this rate. Then the gap gradually increases with the ascending rates, in most cases. We note that the difference in performance is more pronounced for the higher correlation coefficient and the smaller M_1. In particular, in the case of $c = 0.5$, the improvement is around 0.45 dB for $2 \leq R_2 \leq 3$ for both values of M_1. For $M_1 = 4$, the improvement increases as R_2 becomes higher than 3, reaching a peak of 0.75 dB at $R_2 = 0.47$, while for $M_1 = 16$ the performance difference peaks at 0.5 dB. In the case when $c = 0.9$, the improvement over the scheme of (Balasubramanian *et al.*, 1995) when $M_1 = 4$ equals 0.8 dB for $2 \leq R_2 \leq 3$ and gradually increases for $R_2 > 3$, achieving the value of 1.4 dB when $R_2 = 4.4$. For $M_1 = 16$ the performance gain slightly drops, reaching about 0.55 dB for $2 \leq R_2 \leq 3$ and a maximum of 1.1 dB at $R_2 = 4.5$.

c \ R_2	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.4	4.7
0.5	0.003	0.3	0.45	0.45	0.45	0.65	0.55	0.65	0.75
0.9	0.2	0.5	0.8	0.8	0.8	0.95	1.2	1.4	–

Table 4.2: Performance improvement (in dB) at the second decoder over the scheme of (Balasubramanian *et al.*, 1995) for $M_1 = 4$.

c \ R_2	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.1
0.5	–	–	0.45	0.45	0.5	0.5	0.5	0.4	0.4	0.4
0.9	0.15	0.4	0.55	0.45	0.55	0.8	1.0	1.1	0.9	0.9

Table 4.3: Performance improvement (in dB) at the second decoder over the scheme of (Balasubramanian *et al.*, 1995) for $M_1 = 16$.

For a fair comparison we also have to account for the value of D_1, which is shown in Table 4.4. We point out that, for fixed M_1, the value of D_1 obtained with the scheme of (Balasubramanian *et al.*, 1995) is constant, while with our design it varies slightly as R_2 increases up to 3.5, after which it stabilizes. We observe that our scheme outperforms the scheme of (Balasubramanian *et al.*, 1995) at the first decoder when $M_1 = 4$, but it is worse when $M_1 = 16$. However, the loss in the latter case (which is of only 0.1 dB for $R_2 \geq 3.5$) is offset by the gain in performance at decoder 2. Therefore, we conclude that the overall performance of our scheme is higher than that of (Balasubramanian *et al.*, 1995) for both values of M_1. On the other hand, the performance difference tends to decrease as M_1 increases. This is expected since the asymptotic analysis performed in (Balasubramanian *et al.*, 1995) becomes accurate when the rate approaches infinity.

Figure 4.6 illustrates the encoder partitions for the proposed FR-SSQ and for the scheme of (Balasubramanian *et al.*, 1995) when $M_1 = 3$ and $R_2 \approx 2.17$. The figure additionally shows the contour of the joint pdf f_{XY}. It can be noticed that the quantizer of X for the proposed FR-SSQ (Figure 4.6a) has more dense outputs

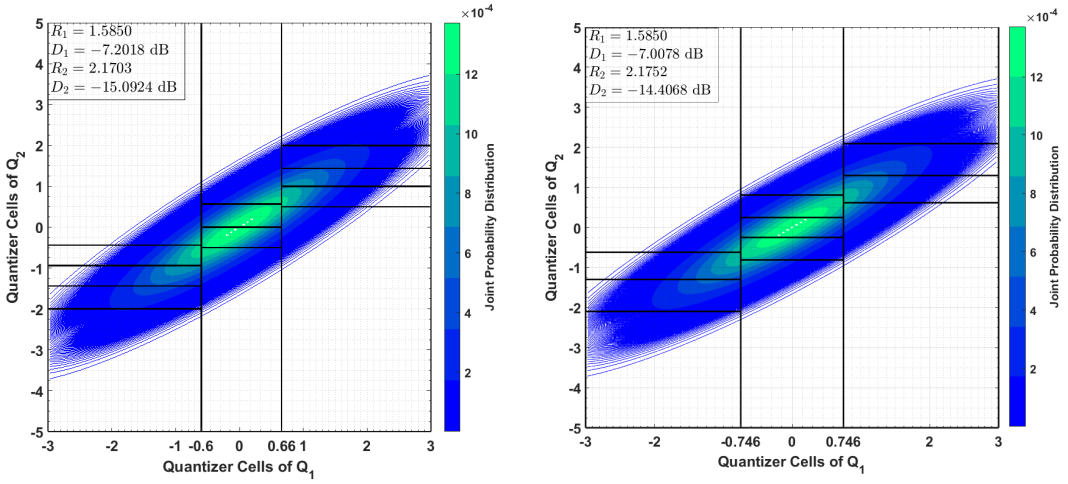
D_1 \ M_1	4		16	
(Balasubramanian *et al.*, 1995)	-9.05		-20.08	
Proposed	$R_2 < 3.5$	$R_2 \geq 3.5$	$R_2 < 3.3$	$R_2 \geq 3.3$
$c = 0.5$	$[-9.30, -9.21]$	-9.30	$[-19.99, -19.89]$	-19.99
Proposed	$R_2 < 2.3$	$R_2 \geq 2.3$	$R_2 < 3.5$	$R_2 \geq 3.5$
$c = 0.9$	$[-9.30, -9.18]$	-9.30	$[-20.07, -19.74]$	-19.99

Table 4.4: Comparison of D_1 between the proposed FR-SSQ and the scheme of (Balasubramanian *et al.*, 1995) for $M_1 = 4$ and 16. The distortion is listed in dB.

in the region where the marginal pdf f_X takes on large values, compared with the counterpart of (Balasubramanian *et al.*, 1995) (Figure 4.6b). This could explain the performance improvement of around 0.19 dB in terms of D_1 for our scheme.

It is instructive to examine the probabilities of the cells of quantizer of X. For the proposed FR-SSQ, we have $P(C_1) = 0.2743$, $P(C_2) = 0.4711$ and $P(C_3) = 0.2546$, while for the scheme of (Balasubramanian *et al.*, 1995), we have $P(C_1) = 0.2278$, $P(C_2) = 0.5444$ and $P(C_3) = 0.2278$. Note that in both cases $P(C_2)$ is higher than $P(C_1)$ and than $P(C_3)$, but cell C_2 is narrower in our design, making its contribution to distortion D_1 smaller than for the scheme of (Balasubramanian *et al.*, 1995). It turns out that this decrease in the distortion of cell C_2 offsets the resulting increase in the distortion of cells C_1 and C_3, thus leading to a smaller value of D_1 for our design.

It can also be observed from Figure 4.6 that in our scheme $M_{2,1} = M_{2,3} > M_{2,2}$, while the opposite holds for the design of (Balasubramanian *et al.*, 1995). This can be attributed to the different constraints imposed in the two designs. Namely, this chapter constraints the average rate at encoder 2, which is $\sum_{i=1}^{M_1} P(C_i) \log_2 M_{2,i}$, to be fixed, while (Balasubramanian *et al.*, 1995) constraints the total number of cells for all encoder 2 quantizers, to be fixed. Since $P(C_1)$ and $P(C_3)$ are lower than $P(C_2)$, our design allows for values of $M_{2,1}$ and $M_{2,3}$ higher than $M_{2,2}$ since an extra cell in either $M_{2,1}$ or $M_{2,3}$ contributes much less to the average rate than an extra cell



(a) Proposed FR-SSQ. (b) Scheme of (Balasubramanian *et al.*, 1995).

Figure 4.6: Encoder partitions for the proposed FR-SSQ (a) and for the scheme of (Balasubramanian *et al.*, 1995) (b) at rate $R_1 = 1.5850$ (i.e., $M_1 = 3$).

in $M_{2,2}$. On the other hand, for the design of (Balasubramanian *et al.*, 1995), an extra cell in any quantizer at encoder 2 has the same effect with respect to meeting the constraint. Therefore, more cells are allocated to $M_{2,2}$ since its distortion has a higher weight in the average distortion D_2 than $M_{2,1}$ or $M_{2,3}$ (i.e., $P(C_2) > P(C_1)$ and $P(C_2) > P(C_3)$). This difference in constraints leads to more quantizer cells being allocated overall at encoder 2 in our design (i.e., 14 cells) than in the competitor scheme (13 cells), ultimately leading to an 0.68 dB improvement in terms of D_2 for the proposed scheme.

4.6 Conclusion

This chapter develops optimal design algorithms for SSQs with convex cells for finite-alphabet correlated sources. Both the FR and EC cases are considered. The cost to be minimized is a weighted sum of distortions and rates. The proposed solutions rely on solving the MWP problem in the EC case, respectively, a length-constrained

MWP problem in the FR case, in a series of WDAGs. The time complexity of each solution is $O(K_1^2 K_2^2)$, where K_1 and K_2 denote the respective cardinalities of the alphabets of the two sources. We also prove that, if the proposed algorithms are applied to fine, uniform discretizations of sources with continuous joint pdf, the performance approaches that of the optimal EC-SSQ, respectively FR-SSQ, with convex cells for the original sources as the discretization becomes more accurate. Experimental results for correlated Gaussian sources corroborate the aforementioned theoretical result.

Chapter 5

Conclusion

This thesis addresses the design of scalar-based quantizers for two-dimensional vectors. In particular, the designs of unrestricted polar quantizer (UPQ) and successively refinable UPQ (SRUPQ) for bivariate circularly symmetric sources in polar coordinates are presented. Moreover, the design algorithm of sequential scalar quantizer (SSQ) for source vectors with correlated elements in Cartesian coordinates is also investigated. Both the entropy-constrained (EC) and fixed-rate (FR) cases are considered in this thesis.

Chapter 2 proposes the globally optimal designs of ECUPQ and FRUPQ, for continuous bivariate circularly symmetric sources. The optimality holds when the magnitude quantizer thresholds are confined to a predefined finite, uniform set. The design of ECUPQ is based on solving a single MWP problem in a WDAG, along with an efficient algorithm finding the number of phase regions for each possible magnitude quantization level. The proposed solution to FRUPQ design is based on dynamic programming expedited by a monotonicity property of the objective function. The time complexity of the proposed ECUPQ and FRUPQ amounts to $O(K^2 + KP_{max})$ and $O(KN^2)$, respectively, where K is the size of the set of possible magnitude

thresholds, P_{max} is the maximum number of phase levels, and N is the number of target qunatizer levels. Both the proposed ECUPQ and FRUPQ outperform the prior practical designs.

Chapter 3 investigates the optimal designs of EC-SRUPQ and FR-SRUPQ. The proposed algorithms are globally optimal when the magnitude quantizers' thresholds are confined to finite, uniform sets. Note that the solution to SRUPQ is more complicated compared with the single-description UPQ design in Chapter 2, as we need to solve the MWP problems (or dynamic programming problems) for multiple node pairs in some other WDAG, where the algorithms in Chapter 2 can no longer be utilized. The asymptotical time complexity is $O(K_1K_2^2P_{cmax})$ for the EC case, where K_1 and K_2 are the sizes of the sets of possible magnitude thresholds of the coarse UPQ and the refined UPQ, respectively, while P_{cmax} is the maximum number of phase levels in any phase quantizer of the coarse UPQ. The time complexity amounts to $O(K_1K_2N'^2N_1)$ for the FR case, where N_1 is the number of bins of the coarse UPQ, while N' denotes the ratio between the number of bins of the fine UPQ and the coarse UPQ. The experimental results show that the performance of the proposed algorithm is close to the corresponding theoretical bounds.

Chapter 4 considers the SSQ design for finite-alphabet sources in Cartesian coordinates, for the case where the components of a two-dimensional vector are correlated. The global optimality holds for the class of SSQs with convex cells. The proposed algorithms of EC-SSQ and FR-SSQ involve solving the MWP and a length-constrained MWP in a series of WDAGs, respectively. The overall running time of each solution is $O(K_1^2K_2^2)$, where K_1 and K_2 denote the respective cardinalities of the alphabets of the two sources. Moreover, we prove that if the proposed algorithms are applied to finite and uniform discretizations of sources with continuous joint pdf, the performance approaches that of the optimal EC-SSQ, respectively FR-SSQ, with convex cells for

the original sources as the discretization becomes more accurate. The proposed EC-SSQ algorithm performs close to the theoretical bounds, while practical performance of FR-SSQ is superior than the prior FR-SSQ design based on high-rate quantization theory.

This thesis addresses the designs of UPQ, SRUPQ and SSQ only for the two-dimensional case, and thus future work may involve the design for high-dimensional sources in both polar and Cartesian coordinates. In Cartesian coordinates, note that for T-dimensional source vector $X^T = (X_1, \cdots, X_T)$ (the samples take value from the same alphabet), the zero delay coding (Yuksel, 2013; Linder and Yüksel, 2014; Wood *et al.*, 2017) allows the quantizer to encode a source sample X_t immediately when it is observed, instead of waiting for grouping T source samples together. Therefore, zero delay coding is practical in real-time processing scenarios, e.g., in real-time streaming systems (Draper *et al.*, 2014; Etezadi *et al.*, 2014) and sensor networks (Akyildiz *et al.*, 2002).

Currently, the literature on zero delay coding mostly focuses on the existence and structure of the optimal quantization policies for Markov sources, e.g., (Witsenhausen, 1979; Walrand and Varaiya, 1983; Yuksel, 2013; Linder and Yüksel, 2014; Wood *et al.*, 2017). Specifically, the authors of (Linder and Yüksel, 2014) prove the existence of optimal zero delay quantizers with convex codecells, for finite T. Therefore, it is worth investigating the practical design of zero delay coding using scalar quantizers with convex cells, potentially and possibly by extending the proposed SSQ framework to high-dimensional vectors ($T \geq 2$).

On the other hand, the SRUPQ and successively refinable SSQ for correlated sources, with general number of refinement stages are also worth further exploration. Another interesting direction to be investigated is the rate of convergence of the approximation error in Theorem 4.1, in terms of B and K. Besides that, the asymptotic

analysis of the EC-SSQ performance is also interesting to address.

Appendix A

Appendix

In this appendix we explain the relation between the formulation of the optimal quantizer design problem as a constrained optimization problem and the corresponding Lagrangian relaxation.

Generally, for any $R \geq 0$, the operational distortion-rate function for a class of quantizers $\mathcal{Q}$ can be defined by

$$D_{\mathcal{Q}}(R) \triangleq \inf_{Q \in \mathcal{Q}} \{D(Q) : R(Q) \leq R\},$$

where $D(Q)$ is the distortion of the quantizer Q, while $R(Q)$ denotes its rate. The infimum is taken over all quantizers in $\mathcal{Q}$ whose rate is not greater than the given rate R. If there is no such a Q with finite distortion and rate $R(Q) \leq R$, then we let $D_{\mathcal{Q}}(R) = +\infty$. Additionally, any quantizer Q achieving $D_{\mathcal{Q}}(R)$ in the sense that $R(Q) \leq R$ and $D(Q) = D_{\mathcal{Q}}(R)$ is called an optimal quantizer (Gyorgy *et al.*, 2003).

In light of this, the quantizer design problem can be formulated as the following

constrained optimization problem

$$\begin{aligned} \min_{Q\in\mathbb{Q}} \quad & D(Q) \\ \text{subject to} \quad & R(Q) \leq R. \end{aligned} \tag{A.1}$$

However, the Lagrangian relaxation is commonly used in prior work (Chou *et al.*, 1989; Muresan and Effros, 2008; Vafin and Kleijn, 2005; Fleming *et al.*, 2004), i.e.,

$$\min_{Q\in\mathbb{Q}} \quad \mathcal{L}(\lambda, Q), \tag{A.2}$$

for some fixed $\lambda > 0$, where $\mathcal{L}(\lambda, Q) \triangleq D(Q) + \lambda R(Q)$.

Note that according to [Theorem 1, (Everett III, 1963)] any quantizer Q achieving the minimum of problem (A.2) with some rate R^*, is also a solution (an optimal quantizer) to problem (A.1) with $R = R^*$. Conversely, any optimal quantizer of problem (A.1) which lies on the lower boundary of the convex hull of $D_{\mathbb{Q}}(R)$, can be found by solving problem (A.2) for some multiplier $\lambda > 0$ [Proposition 3, Chapter 11.9, (Luenberger, 1997)].

In general, the operational distortion-rate function $D_{\mathbb{Q}}(R)$ is not necessarily convex (Chou *et al.*, 1989), and thus the Lagrangian relaxation is not guaranteed to find $D_{\mathbb{Q}}(R)$, for every $R \geq 0$. Actually, whenever the Lagrangian relaxation is utilized, the purpose is to find the lower boundary of the convex hull of $D_{\mathbb{Q}}(R)$ (where $-\lambda$ is the slope of the line supporting the convex hull), and to find the quantizer achieving some point on the lower convex hull. These quantizers are in some sense optimal, and can be used in practice.

The discussion above is for the quantizer design with single encoder-decoder pair, while multiple encoder-decoder pairs may be needed in other systems. In this case,

let us consider another class of quantizer systems $\mathcal{Q}_M$ with M encoder-decoder pairs, where $\mathbf{Q}^M = \{Q_1, Q_2, \cdots, Q_M\} \in \mathcal{Q}_M$, and the distortion and rate of quantizer Q_i are denoted by $D(Q_i)$ and $R(Q_i)$, respectively.

The corresponding quantizer system design problem can be formulated as the following constrained optimization problem

$$\begin{aligned} \min_{\mathbf{Q}^M \in \mathcal{Q}_M} \quad & D(Q_1) \\ \text{subject to} \quad & R(Q_i) \leq R_i, \quad 1 \leq i \leq M, \\ & D(Q_i) \leq D_i, \quad 2 \leq i \leq M. \end{aligned} \tag{A.3}$$

where R_i and D_i are the target rate and distortion for Q_i, respectively.

Next, the Lagrangian relaxation is expressed by

$$\min_{\mathbf{Q}^M \in \mathcal{Q}_M} \quad \mathcal{L}(\boldsymbol{\lambda}, \mathbf{Q}^M), \tag{A.4}$$

for some fixed positive vector $\boldsymbol{\lambda} = \{\lambda_i\}_{i=1}^M$, where $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{Q}^M) \triangleq \sum_{i=1}^M (\rho_i D(Q_i) + \lambda_i R(Q_i))$.

We point out that the formulation of the optimization problem as a minimization of a weighted sum of distortions and rates is also considered in prior work, e.g., (Chou *et al.*, 1989; Muresan and Effros, 2008; Fleming *et al.*, 2004).

Let $\mathcal{RD}$ denote the set of all $2M$-tuples $(R(Q_i), D(Q_i))$ over all quantizers in $\mathcal{Q}_M$, i.e., $\mathcal{RD} = \{(R(Q_i), D(Q_i))_{i=1}^M | \mathbf{Q}^M \in \mathcal{Q}_M\}$. Then any point on the lower boundary of the convex hull of $\mathcal{RD}$ is optimal in some sense, as in the previous discussion. Any such point is the solution of the minimization of a weighted sum of the distortions and rates $\sum_{i=1}^M (\rho_i D(Q_i) + \lambda_i R(Q_i))$, for some choice of positive weight vectors $\boldsymbol{\rho}$ and $\boldsymbol{\lambda}$. Notice that the solution of the minimization problem remains the same if all the

weights are divided by $\sum_{i=1}^{M} \rho_i$.

Note that according to [Theorem 1, (Everett III, 1963)] any solution achieving the minimum of problem (A.4) with $\mathbf{Q}^{M*} = \{Q_1^*, \cdots, Q_M^*\}$, is also a solution to problem (A.3) with $D_i = D(Q_i^*)$, $2 \leq i \leq M$ and $R_i = R(Q_i^*)$, $1 \leq i \leq M$.

Appendix B

Appendix

In this appendix we present the proofs of the claims corresponding to relations (2.15) and (2.16), Lemma 2.1 and Proposition 2.1 in Chapter 2.

Proof of claims corresponding to relations (2.15) and (2.16): Let $P^* \in \mathcal{P}$, and let $\mathcal{L}$ denote the line passing through the point $S(P^*)$ of slope $-\mu$. Consider the following half plane bounded by $\mathcal{L}$,

$$\mathcal{H} \triangleq \{(x, y) | y - f(P^*) + \mu(x - g(P^*)) \geq 0\}.$$

It is clear now that P^* is a solution of problem (2.13), if and only if $\mathcal{U} \subseteq \mathcal{H}$ (i.e., $S(P) \in \mathcal{H}$ for every $P \in \mathcal{P}$). The fact that $\mathcal{U} \subseteq \mathcal{H}$ and that $\mathcal{U} \cap \mathcal{L}$ is nonempty (since it contains $S(P^*)$) means that $\mathcal{L}$ is a support line for $\mathcal{U}$. In other words, we have proved that P^* is a solution to problem (2.13) if and only if the line of slope $-\mu$ passing through $S(P^*)$ is a support line to $\mathcal{U}$.

Now we will prove that if P^* is a solution to problem (2.13), then relation (2.16) holds. Consider $P_1, P_2 \in \mathcal{P}$ such that the segment P_1P^* is the convex hull edge to the

left of P^*, and P^*P_2 is the convex hull edge to the right of P^*. Then we obtain

$$f(P^*) + \mu g(P^*) \leq f(P_1) + \mu g(P_1),$$
$$f(P^*) + \mu g(P^*) \leq f(P_2) + \mu g(P_2).$$

Since function $g(\cdot)$ is increasing on $\mathcal{P}$, after some algebraic manipulations, we further obtain

$$left\_slope(P^*) = \frac{f(P^*) - f(P_1)}{g(P^*) - g(P_1)} \leq -\mu \leq \frac{f(P_2) - f(P^*)}{g(P_2) - g(P^*)} = right\_slope(P^*),$$

which proves relation (2.16).

In order to prove relation (2.15), let us consider a point P_0 in $\hat{\mathcal{U}}$. Then there must exist a support line passing through $S(P_0)$, such that all points $S(P)$ are above or on this line. Let us denote its slope by $-\mu$ (note that we do not need to assume now that $\mu > 0$). Using similar arguments as in the previous discussion, it follows that P_0 is a solution to problem (2.13). Then relation (2.16) holds for P_0 in place of P, which implies that relation (2.15) holds.

Proof of Lemma 2.1:

In light of the discussion above Lemma 2.1, if a point P satisfying (2.17) exists, then the smallest such point equals $P^*_{[a_u,a_v)}$. Then it is sufficient to show the existence of such a point. Since $\lambda > 0$, we have $-\frac{\lambda}{(x[a_u,a_v))^2 \ln 2} < 0$. Thus, it is sufficient to show that $\lim_{P\to\infty} right\_slope(P) = 0$.

Note that if $x[a_u, a_v) = 0$, we obtain $-\frac{\lambda}{(x[a_u,a_v))^2 \ln 2} = -\infty$, then relation (2.17) holds for $P = 1$. Therefore, in the sequel, only the case $x[a_u.a_v) > 0$ is considered.

In view of Proposition 2.1, for every $P \in \mathbb{Z}_+ \setminus \{2\}$, we have $P \in \hat{\mathcal{P}}$. According to

the definition of $right\_slope(P)$, we obtain

$$\begin{aligned}\lim_{P\to\infty} right\_slope(P) &= \lim_{P\to\infty} \frac{f(P+1)-f(P)}{g(P+1)-g(P)} \\ &= \lim_{P\to\infty} \frac{(sinc(\frac{1}{P})+sinc(\frac{1}{P+1}))(sinc(\frac{1}{P})-sinc(\frac{1}{P+1}))}{\ln(P+1)-\ln P} \\ &= -2\lim_{P\to\infty} \frac{sinc(\frac{1}{P+1})-sinc(\frac{1}{P})}{\ln(P+1)-\ln P},\end{aligned}$$

where the last equality is based on the fact that $\lim_{P\to\infty}(sinc(\frac{1}{P})+sinc(\frac{1}{P+1}))=2$. Further, in light of the Cauchy's mean value theorem, since functions $sinc(\frac{1}{P})$ and $\ln P$ are both continuous on $[P, P+1]$, and differentiable on the open interval $(P, P+1)$, there exists some $Q\in(P, P+1)$, such that

$$\frac{sinc(\frac{1}{P+1})-sinc(\frac{1}{P})}{\ln(P+1)-\ln P} = \frac{(sinc(\frac{1}{Q}))'}{(\ln Q)'}.$$

Therefore, it is sufficient to prove that $\lim_{Q\to\infty}\frac{(sinc(\frac{1}{Q}))'}{(\ln Q)'}=0$. For this, note that we have the following sequence of relations

$$\begin{aligned}\lim_{Q\to\infty}\frac{(sinc(\frac{1}{Q}))'}{(\ln Q)'} &= \lim_{Q\to\infty}\frac{\frac{1}{\pi}\sin(\frac{\pi}{Q})-\frac{1}{Q}\cos(\frac{\pi}{Q})}{\frac{1}{Q}} \\ &= \lim_{Q\to\infty}\frac{\pi}{Q}\sin(\frac{\pi}{Q}) \\ &= 0,\end{aligned}$$

where the second equality follows from the L'Hopital's rule. Thus Lemma 2.1 follows.

Proof of Proposition 2.1:

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

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

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.

Appendix C

Appendix

In this appendix we present the proofs of Proposition 3.3 and Proposition 3.4 in Chapter 3. In order to prove Proposition 3.3, we need the following lemmas.

Lemma C.1: For any two points A and B in the plane we use the notation $slope(AB)$ for the slope of the line connecting A and B. Let $P_i \in \hat{\mathcal{P}}$ for $1 \leq i \leq 4$, such that $P_1 < P_2$, $P_3 < P_4$, $P_1 \leq P_3$ and $P_2 \leq P_4$. Then the following holds

$$slope(S(P_1)S(P_2)) \leq slope(S(P_3)S(P_4)). \tag{C.5}$$

Proof: Consider the function $t : [0, \infty) \to \mathbb{R}$ such that for each $x \geq 0$ the pair $(x, t(x))$ is the unique point with abscissa x situated on the lower convex hull of $\mathcal{U}$. Then function t is convex. For each $1 \leq i \leq 4$ let $x_i = g(P_i)$. Then $f(P_i) = t(x_i)$ and the claim follows in virtue of the following lemma.

Lemma C.2: Let $t : \mathbb{R} \to \mathbb{R}$ be a convex function and $x_1 < x_2$, $x_3 < x_4$, $x_1 \leq x_3$ and $x_2 \leq x_4$. Then the following holds

$$\frac{t(x_2) - t(x_1)}{x_2 - x_1} \leq \frac{t(x_4) - t(x_3)}{x_4 - x_3}. \tag{C.6}$$

Proof: Let us assume that $x_2 \neq x_3$. In order to prove (C.6) we will prove the following two inequalities

$$\frac{t(x_2) - t(x_1)}{x_2 - x_1} \leq \frac{t(x_3) - t(x_2)}{x_3 - x_2}, \tag{C.7}$$

$$\frac{t(x_3) - t(x_2)}{x_3 - x_2} \leq \frac{t(x_4) - t(x_3)}{x_4 - x_3}. \tag{C.8}$$

To prove (C.7) we will consider separately the cases 1) $x_2 < x_3$ and 2) $x_3 < x_2$. First note that by performing some algebraic manipulations, (C.7) becomes

$$\frac{1}{(x_2 - x_1)(x_3 - x_2)} \left(t(x_2)(x_3 - x_1) - t(x_1)(x_3 - x_2) - t(x_3)(x_2 - x_1)\right) \leq 0. \tag{C.9}$$

In case 1) one has $x_1 < x_2 < x_3$. Thus $(x_2 - x_1)(x_3 - x_2) > 0$ and (C.9) becomes equivalent to

$$t(x_2)(x_3 - x_1) - t(x_1)(x_3 - x_2) - t(x_3)(x_2 - x_1) \leq 0, \tag{C.10}$$

which is further equivalent to

$$t(x_2) \leq t(x_1)\frac{x_3 - x_2}{x_3 - x_1} + t(x_3)\frac{x_2 - x_1}{x_3 - x_1}. \tag{C.11}$$

Denote $\rho = \frac{x_3 - x_2}{x_3 - x_1}$. Then $0 < \rho < 1$, $\frac{x_2 - x_1}{x_3 - x_1} = 1 - \rho$ and $x_2 = \rho x_1 + (1 - \rho)x_3$. Thus, (C.11) is equivalent to

$$t(\rho x_1 + (1 - \rho)x_3) \leq \rho t(x_1) + (1 - \rho)t(x_3), \tag{C.12}$$

which is true in virtue of the convexity of function t.

Let us consider now case 2). Then $x_1 \leq x_3 < x_2$. If $x_1 = x_3$ then (C.7) holds trivially with equality. Assume now that $x_1 < x_3$. Then $(x_2 - x_1)(x_3 - x_2) < 0$ and

(C.9) becomes equivalent to

$$t(x_2)(x_3 - x_1) + t(x_1)(x_2 - x_3) - t(x_3)(x_2 - x_1) \geq 0, \tag{C.13}$$

which is equivalent to

$$t(x_3) \leq t(x_1)\frac{x_2 - x_3}{x_2 - x_1} + t(x_2)\frac{x_3 - x_1}{x_2 - x_1}. \tag{C.14}$$

If we let $\rho = \frac{x_2 - x_3}{x_2 - x_1}$ then $0 < \rho < 1$ and inequality (C.14) is equivalent to

$$t(\rho x_1 + (1 - \rho)x_2) \leq \rho t(x_1) + (1 - \rho)t(x_2), \tag{C.15}$$

which holds since t is convex. With this observation the proof of (C.7) is complete. The proof of (C.8) follows along similar lines. Clearly, (C.7) and (C.8) further imply (C.6). In the case when $x_2 = x_3$ the proof of (C.6) is analogous to the proof of (C.7) in case 1). These considerations complete the proof of the lemma.

Lemma C.3: $slope(S(1)S(3)) \leq slope(S(2)S(4))$.

Proof: By using the definition of $S(P)$, after some algebraic manipulations we obtain that the above inequality is equivalent to $\frac{-27}{4\pi^2 \ln 3} \leq \frac{-4}{\pi^2 \ln 2}$. This is further equivalent to $27 \ln 2 \geq 16 \ln 3$. By applying the exponential function this becomes equivalent to $2^{27} \geq 3^{16}$. The latter relation is true since $2^{27} = (2^5)^5 \times 2^2$, $3^{16} = (3^3)^5 \times 3$, while $2^5 > 3^3$ and $2^2 > 3$.

Proof of Proposition 3.3:

Let $\delta = \frac{\lambda_2}{(1-\rho)x([b_{K_2}, b_{K_2+1}))^2 \ln 2}$, and let P^* denote $P'_{1,max}$ and P^*_P denote $P'_{P,max}$. Assume first that $P^* \geq 3$. Then, according to Proposition 3.1, $P^* + 1 \in \hat{\mathcal{P}}$ and based

on Lemma 2.1 the following holds

$$-\delta \leq slope(S(P^*)S(P^*+1)). \tag{C.16}$$

Note that $P^* \leq P\lceil\frac{P^*}{P}\rceil$ and $P^*+1 \leq P\lceil\frac{P^*}{P}\rceil + P$ and, based on Proposition 3.1, $P^*, P\lceil\frac{P^*}{P}\rceil$,$P^*+1$ and $P\lceil\frac{P^*}{P}\rceil + P$ are in $\hat{\mathcal{P}}$. Thus, we can apply Lemma C.1 with $P_1 = P^*$, $P_2 = P^*+1$, $P_3 = P\lceil\frac{P^*}{P}\rceil$ and $P_4 = P\lceil\frac{P^*}{P}\rceil + P$ and obtain

$$slope(S(P^*)S(P^*+1)) \leq slope(S(P\lceil\frac{P^*}{P}\rceil)S(P\lceil\frac{P^*}{P}\rceil + P)).$$

The above equation together with (C.16) implies that

$$-\delta \leq slope(S(P\lceil\frac{P^*}{P}\rceil)S(P\lceil\frac{P^*}{P}\rceil + P)). \tag{C.17}$$

Recall that P_P^* is the smallest integer in $\hat{\mathcal{P}}_P$ such that

$$-\delta \leq right\_slope_P(P_P^*) = slope(S(PP_P^*)S(PP_P^* + P)). \tag{C.18}$$

Corroborating the above observation with relation (C.17) and with the fact that $\lceil\frac{P^*}{P}\rceil \in \hat{\mathcal{P}}_P$ (since $\hat{\mathcal{P}}_P = \mathbb{Z}_+$ by Proposition 3.1) and that the slopes of the convex hull of $\mathcal{U}_P$ increase from left to right, we conclude that $P_P^* \leq \lceil\frac{P^*}{P}\rceil < \frac{P^*}{P} + 1$.

It remains to consider now the case when $P^* = 1$. Then relation (C.16) has to be replaced by $-\delta \leq slope(S(1)S(3))$. Assume now that $P \geq 3$. We can apply Lemma C.1 with $P_1 = 1$, $P_2 = 3$, $P_3 = P$ and $P_4 = 2P$ and obtain that $slope(S(1)S(3)) \leq slope(S(P)S(2P))$. This implies that $-\delta \leq slope(S(P \cdot 1)S(P \cdot 2))$. Using further (C.18) we conclude that $P_P^* \leq 1 < \frac{P^*}{P} + 1$. Consider now $P = 2$. Since $P \notin \hat{\mathcal{P}}$ we can no longer apply Lemma C.1 as above. However, we still obtain $slope(S(1)S(3)) \leq$

$slope(S(P\cdot 1)S(P\cdot 2))$ according to Lemma C.3. Then we conclude as above that $P_P^* = 1 < \frac{P^*}{P} + 1$. Thus, the proof is complete.

In order to prove Proposition 3.4, we need the following lemma.

Lemma C.4: Consider $P \in \mathbb{Z}_+$ and let P^* denote the solution to problem (3.20). Then for any $P_1, P_2 \in \hat{\mathcal{P}}_P$ such that $P^* \leq P_1 < P_2$ one has

$$f(PP_1) + \delta g(PP_1) \leq f(PP_2) + \delta g(PP_2). \tag{C.19}$$

Proof: Note that since $g(PP_1) < g(PP_2)$, the above inequality is equivalent (after some algebraic manipulations) to

$$-\delta \leq slope(S(PP_1)S(PP_2)). \tag{C.20}$$

Since $P^*, P_1, P_2 \in \hat{\mathcal{P}}_P$ and $P^* \leq P_1 < P_2$, an argument similar to the proof of Lemma C.1 implies that $right\_slope_P(P^*) \leq slope(S(PP_1)S(PP_2))$. The definition of P^* leads that $-\delta \leq right\_slope_P(P^*)$. Combining the last two inequalities proves relation (C.20). This completes the proof.

Proof of Proposition 3.4:

It is sufficient to prove that if an EC-SRUPQ has $P_i > P_{cmax}$ for some i, then by replacing P_i by P_{cmax} the cost defined in (3.10) does not increase. Note that the portion of the cost affected by P_i is $c(C_i, P_i) = \alpha(C_i, P_i) + \sum_{j=1}^{M_{2,i}} \beta(C_{i,j}, P_i, P_{i,j})$, where

$$\alpha(C_i, P_i) = q(C_i)\rho x^2(C_i)\left(f(P_i) + \frac{\lambda_1}{\rho x^2(C_i)\ln 2} g(P_i)\right), \tag{C.21}$$

$$\beta(C_{i,j}, P_i, P_{i,j}) = q(C_{i,j})(1-\rho)x^2(C_{i,j})\left(f(P_iP_{i,j}) + \frac{\lambda_2}{(1-\rho)x^2(C_{i,j})\ln 2} g(P_iP_{i,j})\right). \tag{C.22}$$

Let P_c^* denote the solution to problem (3.20) for $P = 1$ and $\delta = \frac{\lambda_1}{\rho x^2(C_i)\ln 2}$. Then according to Proposition 2.2 in Section 2.2.3, $P_c^* \leq P''$. Thus, $P_c^* \leq P_{cmax}$. By applying further Lemma C.4 and the fact that $q(C_i)\rho x^2(C_i) > 0$ one obtains that $\alpha(C_i, P_i) \geq \alpha(C_i, P_{cmax})$.

Further, let P_f^* denote the solution to problem (3.20) for $P = P_{i,j}$ and $\delta = \frac{\lambda_2}{(1-\rho)x^2(C_{i,j})\ln 2}$. Then according to Proposition 3.2, one has $P_f^* \leq P'_{P_{i,j},max}$. Using further Proposition 3.3, one obtains $P'_{P_{i,j},max} \leq \frac{P'_{1,max}}{P_{i,j}} + 1 \leq P'_{1,max} + 1$. Since $P_{cmax} \geq P'_{1,max} + 1$ one concludes that $P_f^* \leq P_{cmax}$. By applying Lemma C.4 leads to $\beta(C_{i,j}, P_i, P_{i,j}) \geq \beta(C_{i,j}, P_{cmax}, P_{i,j})$, which concludes the proof.

Appendix D

Appendix

In this appendix we present the proof of Theorem 4.1 in Chapter 4.

Proof of Theorem 4.1:

We will only prove relation (4.22) since (4.23) follows similarly. According to the definition of $\mathcal{F}^*_{EC}$ in Section 4.4, for every $\epsilon > 0$, there exists an EC-SSQ $\mathbf{Q}^*_\epsilon$ such that

$$\mathcal{F}^*_{EC} \leq \mathcal{F}_{EC}(\mathbf{Q}^*_\epsilon, X, Y) \leq \mathcal{F}^*_{EC} + \epsilon. \tag{D.23}$$

Let $\alpha : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a function such that $\mathbb{E}[\alpha(X,Y)]$ is finite. For any $B > 0$ denote $P_B \triangleq \mathbb{P}[X, Y \in [-B, B]]$. Then the following sequence of equalities holds.

$$\begin{aligned}
\mathbb{E}[\alpha(X,Y)] &= \int_{\mathbb{R}} \int_{\mathbb{R}} \alpha(x,y) f_{XY}(x,y)\, dxdy \\
&= \lim_{B\to\infty} \int_{-B}^{B} \int_{-B}^{B} \alpha(x,y) f_{XY}(x,y)\, dxdy \\
&= \lim_{B\to\infty} P_B \int_{-B}^{B} \int_{-B}^{B} \alpha(x,y) \frac{f_{XY}(x,y)}{P_B}\, dxdy \\
&= \lim_{B\to\infty} P_B \mathbb{E}[\alpha(X_B, Y_B)],
\end{aligned}$$

where the last equality is based on the fact that $f_{X_B Y_B}(x,y) = \frac{f_{XY}(x,y)}{P_B}$ for $(x,y) \in$

$[-B, B] \times [-B, B]$ and $f_{X_B Y_B}(x, y) = 0$ for $(x, y) \notin [-B, B] \times [-B, B]$. The above sequence of relations, together with the definition of $\mathcal{F}_{EC}$, implies that

$$\mathcal{F}_{EC}(\mathbf{Q}_\epsilon^*, X, Y) = P_B \mathcal{F}_{EC}(\mathbf{Q}_\epsilon^*, X_B, Y_B) + \epsilon_1(B), \tag{D.24}$$

for some function $\epsilon_1(B)$ such that $\lim_{B\to\infty} \epsilon_1(B) = 0$.

Since $\mathbf{Q}_\epsilon^*$ has convex cells, the partitions of f_1 and $f_{2,i}$, $i \in \mathcal{I}_1$, can be specified by the thresholds separating the cells. According to the specification in Section 4.4, the number of cells within a bounded set is always finite. Thus, the number of thresholds inside a bounded set is also finite.

For each positive real value B and positive integer K, let $\mathbf{Q}_{\epsilon,B,K}^*$ denote the EC-SSQ obtained from $\mathbf{Q}_\epsilon^*$ by restricting the encoding mappings f_1 and $f_{2,i}$ to the interval $[-B, B]$ and by rounding down each threshold in $(-B, B)$ to the closest, no larger, value in $\mathcal{U}_{B,K}$. In other words, if $a \in (-B, B)$ is a threshold of $\mathbf{Q}_\epsilon^*$, then it is rounded down to $t_k^{(B)}$ such that $t_k^{(B)} \leq a < t_{k+1}^{(B)}$. Let us assume that K is large enough so that this rounding operation generates different values for different thresholds in $\mathbf{Q}_\epsilon^*$ and all these values are larger than $-B$. Further, let $\mathbf{v}_B^*$ denote the vector obtained by stacking all thresholds of $\mathbf{Q}_\epsilon^*$ which are in $(-B, B)$ and let $\mathbf{v}_{B,K}^*$ denote the vector obtained by stacking the corresponding thresholds of $\mathbf{Q}_{\epsilon,B,K}^*$ in the same order. Both vectors have the same dimension. Then $\mathcal{F}_{EC}(\mathbf{Q}_\epsilon^*, X_B, Y_B)$ can be regarded as a function of $\mathbf{v}_B^*$, while $\mathcal{F}_{EC}(\mathbf{Q}_{\epsilon,B,K}^*, X_B, Y_B)$ is the same function of $\mathbf{v}_{B,K}^*$. Let β denote this function. Thus,

$$\mathcal{F}_{EC}(\mathbf{Q}_\epsilon^*, X_B, Y_B) = \beta(\mathbf{v}_B^*) \text{ and } \mathcal{F}_{EC}(\mathbf{Q}_{\epsilon,B,K}^*, X_B, Y_B) = \beta(\mathbf{v}_{B,K}^*).$$

It can be easily seen that $\lim_{K\to\infty} \|\mathbf{v}_{B,K}^* - \mathbf{v}_B^*\|^2 = 0$, where $\|\cdot\|$ denotes the Euclidian

norm. Additionally, we point out that β is a continuous function, as stated in Lemma D.1, which is stated and proved at the end of this appendix. Thus, we obtain that

$$\mathcal{F}_{EC}(\mathbf{Q}^*_{\epsilon,B,K}, X_B, Y_B) = \mathcal{F}_{EC}(\mathbf{Q}^*_{\epsilon}, X_B, Y_B) + \delta(B,K), \tag{D.25}$$

for some $\delta(B,K)$ such that $\lim_{K\to\infty} \delta(B,K) = 0$.

Consider now the EC-SSQ $\tilde{\mathbf{Q}}^*_{\epsilon,B,K}$ for the pair of discrete RVs $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$, constructed from $\mathbf{Q}^*_{\epsilon,B,K}$ as explained next. For each cell $C_i = (t_u^{(B)}, t_v^{(B)}]$ of the first encoder of $\mathbf{Q}^*_{\epsilon,B,K}$, the corresponding cell in $\tilde{\mathbf{Q}}^*_{\epsilon,B,K}$ is $\tilde{C}_i = \{x_{u+1}^{(B)}, \ldots, x_v^{(B)}\}$. For each cell $C_{i,j} = (t_m^{(B)}, t_n^{(B)}]$ of the second encoder of $\mathbf{Q}^*_{\epsilon,B,K}$, the corresponding cell in $\tilde{\mathbf{Q}}^*_{\epsilon,B,K}$ is $\tilde{C}_{i,j} = \{y_{m+1}^{(B)}, \ldots, y_n^{(B)}\}$. It follows that $\mathbb{P}[\tilde{X}_{B,K} \in \tilde{C}_i] = \mathbb{P}[X_B \in C_i]$ and $\mathbb{P}[\tilde{X}_{B,K} \in \tilde{C}_i, \tilde{Y}_{B,K} \in \tilde{C}_{i,j}] = \mathbb{P}[X_B \in C_i, Y_B \in C_{i,j}]$ for all i and j. This observation implies that

$$\begin{aligned} R_{EC,1}(\tilde{\mathbf{Q}}^*_{\epsilon,B,K}, \tilde{X}_{B,K}) &= R_{EC,1}(\mathbf{Q}^*_{\epsilon,B,K}, X_B), \\ R_{EC,2}(\tilde{\mathbf{Q}}^*_{\epsilon,B,K}, \tilde{X}_{B,K}, \tilde{Y}_{B,K}) &= R_{EC,2}(\mathbf{Q}^*_{\epsilon,B,K}, X_B, Y_B). \end{aligned} \tag{D.26}$$

Next we will show that the following hold

$$D_1(\tilde{\mathbf{Q}}^*_{\epsilon,B,K}, \tilde{X}_{B,K}) = D_1(\mathbf{Q}^*_{\epsilon,B,K}, X_B) - D_{\tilde{X}_{B,K}}, \tag{D.27}$$

$$D_2(\tilde{\mathbf{Q}}^*_{\epsilon,B,K}, \tilde{X}_{B,K}, \tilde{Y}_{B,K}) = D_2(\mathbf{Q}^*_{\epsilon,B,K}, X_B, Y_B) - \gamma(\mathbf{Q}^*_{\epsilon,B,K}), \tag{D.28}$$

where

$$D_{\tilde{X}_{B,K}} \triangleq \sum_{k=1}^{K} \int_{t_{k-1}^{(B)}}^{t_k^{(B)}} (x - x_k^{(B)})^2 f_{X_B}(x)\, dx,$$

$$\gamma(\mathbf{Q}^*_{\epsilon,B,K}) \triangleq \sum_{i=1}^{M_1}\sum_{l=1}^{K}\int_{t_{l-1}^{(B)}}^{t_l^{(B)}}(y-y_l^{(B)})(y+y_l^{(B)}-2g_2(i,f_2(i,y)))\int_{C_i} f_{X_BY_B}(x,y)\,dxdy.$$

Note that $|y+y_l^{(B)}-2g_2(i,f_2(i,y))|\leq 4B$ and $|y-y_l^{(B)}|\leq\frac{2B}{K}$ when $y\in[t_{l-1}^{(B)},t_l^{(B)}]$. Thus, we obtain that $|\gamma(\mathbf{Q}^*_{\epsilon,B,K})|\leq\frac{8B^2}{K}\sum_{i=1}^{M_1}\sum_{l=1}^{K}\int_{t_{l-1}^{(B)}}^{t_l^{(B)}}\int_{C_i} f_{X_BY_B}(x,y)\,dxdy=\frac{8B^2}{K}$, which leads to

$$\lim_{K\to\infty}\gamma(\mathbf{Q}^*_{\epsilon,B,K})=0. \tag{D.29}$$

In order to prove (D.27) let $C_i=(t_u^{(B)},t_v^{(B)}]$. It follows that

$$\begin{aligned}
&\int_{t_u^{(B)}}^{t_v^{(B)}}(x-g_1(i))^2 f_{X_B}(x)\,dx=\sum_{k=u+1}^{v}\int_{t_{k-1}^{(B)}}^{t_k^{(B)}}(x-x_k^{(B)}+x_k^{(B)}-g_1(i))^2 f_{X_B}(x)\,dx=\\
&\sum_{k=u+1}^{v}\left(\int_{t_{k-1}^{(B)}}^{t_k^{(B)}}(x-x_k^{(B)})^2 f_{X_B}(x)\,dx+\int_{t_{k-1}^{(B)}}^{t_k^{(B)}}(x_k^{(B)}-g_1(i))^2 f_{X_B}(x)\,dx+\right.\\
&\left.2(x_k^{(B)}-g_1(i))\int_{t_{k-1}^{(B)}}^{t_k^{(B)}}(x-x_k^{(B)})f_{X_B}(x)\,dx\right)=\\
&\sum_{k=u+1}^{v}\int_{t_{k-1}^{(B)}}^{t_k^{(B)}}(x-x_k^{(B)})^2 f_{X_B}(x)\,dx+\sum_{k=u+1}^{v}(x_k^{(B)}-g_1(i))^2 p_{\tilde{X}_{B,K}}(x_k^{(B)}),
\end{aligned}$$

where the last equality is due to the fact that $x_k^{(B)}$ is the centroid of $(t_{k-1}^{(B)},t_k^{(B)})$ with respect to $f_{X_B}(x)$ and thus $\int_{t_{k-1}^{(B)}}^{t_k^{(B)}}(x-x_k^{(B)})f_{X_B}(x)\,dx=0$, and that $p_{\tilde{X}_{B,K}}(x_k^{(B)})=\int_{t_{k-1}^{(B)}}^{t_k^{(B)}} f_{X_B}(x)\,dx$. The above observation implies (D.27). In order to prove (D.28), let

$C_{i,j} = (t_m^{(B)}, t_n^{(B)}]$. It follows that

$$\begin{aligned}
&\int_{t_m^{(B)}}^{t_n^{(B)}} (y - g_2(i,j))^2 \int_{C_i} f_{X_B Y_B}(x,y)\ dxdy \\
&= \sum_{l=m+1}^{n} \int_{t_{l-1}^{(B)}}^{t_l^{(B)}} \Big((y - y_l^{(B)})(y + y_l^{(B)} - 2g_2(i,j)) + (y_l^{(B)} - g_2(i,j))^2\Big) \int_{C_i} f_{X_B Y_B}(x,y)\ dxdy \\
&= \sum_{l=m+1}^{n} \int_{t_{l-1}^{(B)}}^{t_l^{(B)}} (y - y_l^{(B)})(y + y_l^{(B)} - 2g_2(i,j)) \int_{C_i} f_{X_B Y_B}(x,y)\ dxdy \\
&\quad + \sum_{l=m+1}^{n} (y_l^{(B)} - g_2(i,j))^2 \mathbb{P}[\tilde{X}_{B,K} \in \tilde{C}_i, \tilde{Y}_{B,K} = y_l^{(B)}],
\end{aligned}$$

where the last equality uses the fact that $\mathbb{P}[\tilde{X}_{B,K} \in \tilde{C}_i, \tilde{Y}_{B,K} = y_l^{(B)}] = \int_{t_{l-1}^{(B)}}^{t_l^{(B)}} \int_{C_i} f_{X_B Y_B}(x,y)\ dxdy$. The above observation implies (D.28). Further, relations (D.26)-(D.28) lead to

$$\mathcal{F}_{EC}(\mathbf{Q}^*_{\epsilon,B,K}, X_B, Y_B) = \mathcal{F}_{EC}(\tilde{\mathbf{Q}}^*_{\epsilon,B,K}, \tilde{X}_{B,K}, \tilde{Y}_{B,K}) + \rho D_{\tilde{X}_{B,K}} + (1-\rho)\gamma(\mathbf{Q}^*_{\epsilon,B,K}). \quad \text{(D.30)}$$

Further, recall that $\hat{\mathbf{Q}}_{B,K}$ is the optimal EC-SSQ (with convex cells) for the pair of RVs $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$. Let $\mathbf{Q}_{B,K}$ be the corresponding EC-SSQ for (X_B, Y_B) with thresholds in $\mathcal{U}_{B,K}$, according to the correspondence described in the paragraph after equation (D.25). Then we have, similarly to (D.30),

$$\mathcal{F}_{EC}(\mathbf{Q}_{B,K}, X_B, Y_B) = \mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K}, \tilde{X}_{B,K}, \tilde{Y}_{B,K}) + \rho D_{\tilde{X}_{B,K}} + (1-\rho)\gamma(\mathbf{Q}_{B,K}). \quad \text{(D.31)}$$

Now consider extending the EC-SSQ $\mathbf{Q}_{B,K}$ to an EC-SSQ $\bar{\mathbf{Q}}_{B,K}$ for X, Y, as follows. The encoding partition for X in $\bar{\mathbf{Q}}_{B,K}$ has two more cells, namely $(-\infty, -B)$ and (B, ∞), both having the mean of X as reconstruction. Likewise, when $X \in [-B, B]$, the encoder for Y has two more cells, namely $(-\infty, -B)$ and (B, ∞), both having the mean of Y as reconstruction. When $X \notin [-B, B]$, the encoder for Y sends only

one symbol and the reconstruction is the mean of Y. It can be readily seen that

$$\mathcal{F}_{EC}(\bar{\mathbf{Q}}_{B,K}, X, Y) = P_B \mathcal{F}_{EC}(\mathbf{Q}_{B,K}, X_B, Y_B) + \epsilon_2(B), \tag{D.32}$$

for some function $\epsilon_2(B)$ such that $\lim_{B\to\infty} \epsilon_2(B) = 0$.

The aforementioned discussion implies the following sequence of relations

$$\begin{aligned}
&\mathcal{F}^*_{EC} \\
&\overset{(a)}{\leq} \quad \mathcal{F}_{EC}(\bar{\mathbf{Q}}_{B,K}, X, Y) \\
&\overset{(b)}{=} \quad P_B \mathcal{F}_{EC}(\mathbf{Q}_{B,K}, X_B, Y_B) + \epsilon_2(B) \\
&\overset{(c)}{=} \quad P_B \left(\mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K}, \tilde{X}_{B,K}, \tilde{Y}_{B,K}) + \rho D_{\tilde{X}_{B,K}} + (1-\rho)\gamma(\mathbf{Q}_{B,K}) \right) + \epsilon_2(B) \\
&\overset{(d)}{\leq} \quad P_B \left(\mathcal{F}_{EC}(\tilde{\mathbf{Q}}^*_{\epsilon,B,K}, \tilde{X}_{B,K}, \tilde{Y}_{B,K}) + \rho D_{\tilde{X}_{B,K}} + (1-\rho)\gamma(\mathbf{Q}_{B,K}) \right) + \epsilon_2(B) \\
&\overset{(e)}{=} \quad P_B \left(\mathcal{F}_{EC}(\mathbf{Q}^*_{\epsilon,B,K}, X_B, Y_B) + (1-\rho)(\gamma(\mathbf{Q}_{B,K}) - \gamma(\mathbf{Q}^*_{\epsilon,B,K})) \right) + \epsilon_2(B) \\
&\overset{(f)}{=} \quad P_B \left(\mathcal{F}_{EC}(\mathbf{Q}^*_{\epsilon}, X_B, Y_B) + \delta(B,K) + (1-\rho)(\gamma(\mathbf{Q}_{B,K}) - \gamma(\mathbf{Q}^*_{\epsilon,B,K})) \right) + \epsilon_2(B) \\
&\overset{(g)}{=} \quad \mathcal{F}_{EC}(\mathbf{Q}^*_{\epsilon}, X, Y) - \epsilon_1(B) + P_B(\delta(B,K) + (1-\rho)(\gamma(\mathbf{Q}_{B,K}) - \gamma(\mathbf{Q}^*_{\epsilon,B,K}))) + \epsilon_2(B) \\
&\overset{(h)}{\leq} \quad \mathcal{F}^*_{EC} + \epsilon - \epsilon_1(B) + P_B(\delta(B,K) + (1-\rho)(\gamma(\mathbf{Q}_{B,K}) - \gamma(\mathbf{Q}^*_{\epsilon,B,K}))) + \epsilon_2(B).
\end{aligned}$$

Notice that (a) follows from the definition of $\mathcal{F}^*_{EC}$, (b) is based on (D.32), (c) follows from (D.31), (d) holds in virtue of the optimality of $\hat{\mathbf{Q}}_{B,K}$ for $(\tilde{X}_{B,K}, \tilde{Y}_{B,K})$, (e) follows from (D.30), (f) from (D.25), (g) is based on (D.24) and (h) is based on (D.23). Next we use the sequence of relations $(a) - (h)$ and apply the fact that $A_1 \leq A_2 \leq A_3$ implies that $A_2 - A_1 \leq A_3 - A_1$ and $A_3 - A_2 \leq A_3 - A_1$, for $A_1 = \mathcal{F}^*_{EC}$, A_2 being the

right hand side of (c) and A_3 being the right hand side of (h). Thus, we obtain

$$\begin{aligned} &P_B\left(\mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K},\tilde{X}_{B,K},\tilde{Y}_{B,K})+\rho D_{\tilde{X}_{B,K}}+(1-\rho)\gamma(\mathbf{Q}_{B,K})\right)+\epsilon_2(B)-\mathcal{F}^*_{EC}\\ &\leq \epsilon-\epsilon_1(B)+P_B(\delta(B,K)+(1-\rho)(\gamma(\mathbf{Q}_{B,K})-\gamma(\mathbf{Q}^*_{\epsilon,B,K})))+\epsilon_2(B). \end{aligned} \tag{D.33}$$

$$\begin{aligned} &\mathcal{F}^*_{EC}+\epsilon-\epsilon_1(B)+P_B(\delta(B,K)+(1-\rho)(\gamma(\mathbf{Q}_{B,K})-\gamma(\mathbf{Q}^*_{\epsilon,B,K}))+\epsilon_2(B)\\ &-P_B\left(\mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K},\tilde{X}_{B,K},\tilde{Y}_{B,K})+\rho D_{\tilde{X}_{B,K}}+(1-\rho)\gamma(\mathbf{Q}_{B,K})\right)-\epsilon_2(B)\leq\\ &\epsilon-\epsilon_1(B)+P_B(\delta(B,K)+(1-\rho)(\gamma(\mathbf{Q}_{B,K})-\gamma(\mathbf{Q}^*_{\epsilon,B,K})))+\epsilon_2(B). \end{aligned} \tag{D.34}$$

Relation (D.33) implies that

$$\mathcal{F}^*_{EC}-P_B\mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K},\tilde{X}_{B,K},\tilde{Y}_{B,K})\geq P_B\left(\rho D_{\tilde{X}_{B,K}}-\delta(B,K)+(1-\rho)\gamma(\mathbf{Q}^*_{\epsilon,B,K})\right)+\epsilon_1(B)-\epsilon.$$

Relation (D.34) leads to

$$\mathcal{F}^*_{EC}-P_B\mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K},\tilde{X}_{B,K},\tilde{Y}_{B,K})\leq P_B\left(\rho D_{\tilde{X}_{B,K}}+(1-\rho)\gamma(\mathbf{Q}_{B,K})\right)+\epsilon_2(B).$$

The above two inequalities together with (D.29) and $\lim_{K\to\infty} D_{\tilde{X}_{B,K}}=\lim_{K\to\infty}\gamma(\mathbf{Q}_{B,K})=\lim_{K\to\infty}\delta(B,K)=0$, $\lim_{B\to\infty}\epsilon_1(B)=\lim_{B\to\infty}\epsilon_2(B)=0$ and $\lim_{B\to\infty}P_B=1$ lead to

$$0\leq\lim_{B\to\infty}\lim_{K\to\infty}\mathcal{F}_{EC}(\hat{\mathbf{Q}}_{B,K},\tilde{X}_{B,K},\tilde{Y}_{B,K})-\mathcal{F}^*_{EC}\leq\epsilon,$$

for every $\epsilon>0$, which implies that relation (4.22) holds. Relation (4.23) follows similarly.

Lemma D.1: Function $\beta(\cdot)$ is continuous as a function of the thresholds of the quantizer.

Proof of Lemma D.1:

We will prove that the function $\beta(\cdot)$ is continuous as a function of thresholds $\mathbf{v}_B^*$ of the quantizer $\mathbf{Q}_\epsilon^*$. The proof is based on the result [a.Theorem, Chapter 9.31, (Shilov *et al.*, 1996)] that if a function $f(x)$ is integrable on a bounded interval $[x_1, x_2]$, then for $x_1 \leq x \leq x_2$, the function $F(x) = \int_{x_1}^{x} f(t)\ dt$ is continuous on $[x_1, x_2]$.

It can be observed that

$$\beta(\mathbf{v}_B^*) = \rho D_1(\mathbf{Q}_\epsilon^*, X_B) + (1-\rho) D_2(\mathbf{Q}_\epsilon^*, X_B, Y_B) + \lambda_1 R_{EC,1}(\mathbf{Q}_\epsilon^*, X_B) + \lambda_2 R_{EC,2}(\mathbf{Q}_\epsilon^*, X_B, Y_B),$$

and thus it is sufficient to prove that each term in $\beta(\mathbf{v}_B^*)$ is continuous as a function of boundaries of each cell. Further, we consider the cells by $C_i = (a, b] \in (-B, B)$ and $C_{i,j} = (c, d] \in (-B, B)$. It then follows that the distortion and entropy of each cell C_i can be represented, respectively by

$$d_1(C_i, X_B) = \int_a^b (x - \hat{x}((a,b])^2 f_{X_B}(x)\ dx,$$
$$r_{EC,1}(C_i, X_B) = -P(C_i) \log_2 P(C_i),$$

where $P(C_i) = \int_a^b f_{X_B}(x)\ dx$ and $P(C_i) > 0$, which follows from the hypothesis that $f_{XY}(x, y) > 0$ for any $x, y \in \mathbb{R}$ in Theorem 4.1. Considering the fact that the marginal pdf $f_{X_B}(x)$ and the function $\log_2(\cdot)$ are both continuous, then $r_{EC,1}(C_i, X_B)$ is continuous in the boundaries of C_i. Further, it is known that the centroid $\hat{x}((a, b])$ is continuous and non-decreasing in a and b (Trushkin, 1982), which implies that $d_1(C_i, X_B)$ is also continuous in a and b. As a consequence, the terms $D_1(\mathbf{Q}_\epsilon^*, X_B)$ and $R_{EC,1}(\mathbf{Q}_\epsilon^*, X_B)$ are continuous as a function of $\mathbf{v}_B^*$.

Similarly, the distortion and entropy of each cell $C_{i,j}$ can be expressed, respectively

by

$$
\begin{aligned}
d_2(C_{i,j}, X_B, Y_B) &= \int_a^b \int_c^d (y - \hat{y}((c,d]|(a,b]))^2 f_{X_B,Y_B}(x,y)\ dy\ dx \\
&= \int_c^d y^2 f_{Y_B}(y)\ dy - (\hat{y}((c,d]|(a,b]))^2 \int_a^b \int_c^d f_{X_B,Y_B}(x,y)\ dy\ dx, \\
r_{EC,2}(C_{i,j}, X_B, Y_B) &= -P(C_{i,j}) \log_2 P(C_{i,j}) + P(C_i) \log_2 P(C_i),
\end{aligned}
$$

where $P(C_{i,j}) = \int_a^b \int_c^d f_{X_B,Y_B}(x,y)\ dy\ dx$ and $P(C_{i,j}) > 0$, since $f_{XY}(x,y) > 0$ as mentioned in the hypothesis in Theorem 4.1. Let us prove first the continuity of $d_2(C_{i,j}, X_B, Y_B)$ as a function of a, b, c and d. Note that the summation $\sum_{i=1}^{M_1} \sum_{j=1}^{M_{2,i}} \int_c^d y^2 f_{Y_B}(y)\ dy$ is the second moment of y, which is a constant. Therefore, we only need to prove the continuity of term $\int_a^b \int_c^d f_{X_B,Y_B}(x,y)\ dy\ dx$ and of the centroid $\hat{y}((c,d]|(a,b])$ as functions a, b, c and d.

Let us denote by $F_1(c,d,x) = \int_c^d f_{X_B,Y_B}(x,y)\ dy$, which is continuous as a function of c and d, since the joint pdf $f_{X_B,Y_B}(x,y)$ is continuous. Consider $F_2(a,b,c,d) = \int_a^b F_1(c,d,x)\ dx$. To prove that $F_2(a,b,c,d)$ is continuous as a function of a and b, it is sufficient to prove that $F_1(c,d,x)$ is continuous as a function of x. In other words, we need to prove that $\lim_{x \to x_0} |F_1(c,d,x) - F_1(c,d,x_0)| = 0$ for every $x_0 \in [a,b]$. For

this note that

$$\begin{aligned}
\lim_{x\to x_0} |F_1(c,d,x) - F_1(c,d,x_0)| &= \lim_{x\to x_0} \left| \int_c^d f_{X_B,Y_B}(x,y)\ dy - \int_c^d f_{X_B,Y_B}(x_0,y)\ dy \right| \\
&= \lim_{x\to x_0} \left| \int_c^d (f_{X_B,Y_B}(x,y) - f_{X_B,Y_B}(x_0,y))\ dy \right| \\
&\leq \lim_{x\to x_0} \int_c^d |f_{X_B,Y_B}(x,y) - f_{X_B,Y_B}(x_0,y)|\ dy \\
&\leq (d-c) \lim_{x\to x_0} \max_{y\in[c,d]} |f_{X_B,Y_B}(x,y) - f_{X_B,Y_B}(x_0,y)| \\
&= (d-c) \lim_{x\to x_0} |f_{X_B,Y_B}(x,y^*) - f_{X_B,Y_B}(x_0,y^*)| \\
&\overset{(i)}{=} 0,
\end{aligned}$$

where $y^* = \arg\max_{y\in[c,d]} |f_{X_B,Y_B}(x,y) - f_{X_B,Y_B}(x_0,y)|$, and relation (ii) is due to the fact that the joint pdf $f_{X_B,Y_B}(x,y)$ is continuous in x (leading to $\lim_{x\to x_0} |f_{X_B,Y_B}(x,y^*) - f_{X_B,Y_B}(x_0,y^*)| = 0$). It is proved now that $F_2(a,b,c,d)$ is continuous in a and b. Since the order of integral in $\int_a^b \int_c^d f_{X_B,Y_B}(x,y)\ dy\ dx$ can be exchanged, the continuity of $F_2(a,b,c,d)$ in c and d also follows.

Further, the proof of continuity of the centroid $\hat{y}((c,d]|(a,b])$ uses similar arguments. Based on the above discussion, it is clear that $D_2(\mathbf{Q}_\epsilon^*, X_B, Y_B)$ is continuous as a function of $\mathbf{v}_B^*$.

Next, let us consider the continuity of $r_{EC,2}(C_{i,j}, X_B, Y_B)$ as a function of a, b, c and d. Now it is known that $P(C_{i,j})$ is continuous using the above result, and then considering the fact that both $\log_2(\cdot)$ and $P(C_i)$ are also continuous, the continuity of $R_{EC,2}(\mathbf{Q}_\epsilon^*, X_B, Y_B)$ as a function of $\mathbf{v}_B^*$ follows.

Bibliography

Aggarwal, A., Klawe, M. M., Moran, S., Shor, P., and Wilber, R. (1987). Geometric applications of a matrix-searching algorithm. *Algorithmica*, **2**(1-4), 195–208.

Aggarwal, A., Schieber, B., and Tokuyama, T. (1994). Finding a minimum-weightk-link path in graphs with the concave monge property and applications. *Discrete & Computational Geometry*, **12**(3), 263–280.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, **38**(4), 393–422.

Balasubramanian, R., Bouman, C. A., and Allebach, J. P. (1994). Sequential scalar quantization of color images. *Journal of Electronic Imaging*, **3**(1), 45–60.

Balasubramanian, R., Bouman, C. A., and Allebach, J. P. (1995). Sequential scalar quantization of vectors: An analysis. *IEEE Transactions on Image Processing*, **4**(9), 1282–1295.

Bruckstein, A. M., Holt, R. J., and Netravali, A. N. (1998). Holographic representations of images. *IEEE Transactions on Image Processing*, **7**(11), 1583–1597.

Brunk, H. and Farvardin, N. (1996). Fixed-rate successively refinable scalar quantizers. In *Proc. Data Compression Conference*, pages 250–259. IEEE.

Bucklew, J. and Gallagher, N. (1979a). Quantization schemes for bivariate gaussian random variables. *IEEE Transactions on Information Theory*, **25**(5), 537–543.

Bucklew, J. and Gallagher, N. (1979b). Two-dimensional quantization of bivariate circularly symmetric densities. *IEEE Transactions on Information Theory*, **25**(6), 667–671.

Burkard, R. E., Klinz, B., and Rudolf, R. (1996). Perspectives of monge properties in optimization. *Discrete Applied Mathematics*, **70**(2), 95–161.

Chang, J. Z. and Allebach, J. P. (1993). Optimal sequential scalar quantization of vectors. In *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, pages 966–971. IEEE.

Chen, J., Dumitrescu, S., Zhang, Y., and Wang, J. (2010). Robust multiresolution coding. *IEEE Transactions on Communications*, **58**(11), 3186–3195.

Chou, P. A., Lookabaugh, T., and Gray, R. M. (1989). Entropy-constrained vector quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **37**(1), 31–42.

Draper, S. C., Chang, C., and Sahai, A. (2014). Lossless coding for distributed streaming sources. *IEEE Transactions on Information Theory*, **60**(3), 1447–1474.

Dumitrescu, S. (2016). On the design of optimal noisy channel scalar quantizer with random index assignment. *IEEE Transactions on Information Theory*, **62**(2), 724–735.

Dumitrescu, S. and Wu, X. (2002). Optimal multiresolution quantization for scalable multimedia coding. In *Proc. Information Theory Workshop*, pages 139–142. IEEE.

Dumitrescu, S. and Wu, X. (2004). Algorithms for optimal multi-resolution quantization. *Journal of Algorithms*, **50**(1), 1–22.

Dumitrescu, S. and Wu, X. (2005). Optimal two-description scalar quantizer design. *Algorithmica*, **41**(4), 269–287.

Dumitrescu, S. and Wu, X. (2007). Lagrangian optimization of two-description scalar quantizers. *IEEE Transactions on Information Theory*, **53**(11), 3990–4012.

Effros, M. and Dugatkin, D. (2004). Multiresolution vector quantization. *IEEE transactions on information theory*, **50**(12), 3130–3145.

Equitz, W. H. and Cover, T. M. (1991). Successive refinement of information. *IEEE Transactions on Information Theory*, **37**(2), 269–275.

Etezadi, F., Khisti, A., and Trott, M. (2014). Zero-delay sequential transmission of markov sources over burst erasure channels. *IEEE Transactions on Information Theory*, **60**(8), 4584–4613.

Everett III, H. (1963). Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, **11**(3), 399–417.

Fleming, M., Zhao, Q., and Effros, M. (2004). Network vector quantization. *IEEE Transactions on Information Theory*, **50**(8), 1584–1604.

Gallagher, N. (1978). Quantizing schemes for the discrete fourier transform of a random time-series. *IEEE Transactions on Information Theory*, **24**(2), 156–163.

Gersho, A. (1979). Asymptotically optimal block quantization. *IEEE Transactions on information theory*, **25**(4), 373–380.

Gish, H. and Pierce, J. (1968). Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, **14**(5), 676–683.

Gray, R. M. and Neuhoff, D. L. (1998). Quantization. *IEEE transactions on information theory*, **44**(6), 2325–2383.

Gyorgy, A. and Linder, T. (2002). On the structure of optimal entropy-constrained scalar quantizers. *IEEE transactions on information theory*, **48**(2), 416–427.

Gyorgy, A., Linder, T., Chou, P. A., and Betts, B. J. (2003). Do optimal entropy-constrained quantizers have a finite or infinite number of codewords? *IEEE Transactions on Information Theory*, **49**(11), 3031–3037.

Jafarkhani, H. and Tarokh, V. (1999). Design of successively refinable trellis-coded quantizers. *IEEE Transactions on Information Theory*, **45**(5), 1490–1497.

Jovanović, A. Ž., Perić, Z. H., Nikolić, J. R., and Dinčić, M. R. (2016). Asymptotic analysis and design of restricted uniform polar quantizer for gaussian sources. *Digital Signal Processing*, **49**, 24–32.

Kingsbury, N. and Reeves, T. (2003). Redundant representation with complex wavelets: How to achieve sparsity. In *Proc. International Conference on Image Processing*, volume 1, pages I–45–48. IEEE.

Kostina, V. and Tuncel, E. (2017). The rate-distortion function for successive refinement of abstract sources. In *Proc. International Symposium on Information Theory*, pages 1923–1927. IEEE.

Linder, T. and Yüksel, S. (2014). On optimal zero-delay coding of vector markov sources. *IEEE Trans. Information Theory*, **60**(10), 5975–5991.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, **28**(2), 129–137.

Luenberger, D. G. (1997). *Optimization by vector space methods*. John Wiley & Sons.

Max, J. (1960). Quantizing for minimum distortion. *IRE Transactions on Information Theory*, **6**(1), 7–12.

Moo, P. W. and Neuhoff, D. L. (1998). Uniform polar quantization revisited. In *Proc. International Symposium on Information Theory*, page 100. IEEE.

Muresan, D. and Effros, M. (2002). Quantization as histogram segmentation: globally optimal scalar quantizer design in network systems. In *Proc. Data Compression Conference*, pages 302–311. IEEE.

Muresan, D. and Effros, M. (2008). Quantization as histogram segmentation: Optimal scalar quantizer design in network systems. *IEEE Transactions on Information Theory*, **54**(1), 344–366.

Nazari, P., Chun, B.-K., Tzeng, F., and Heydari, P. (2014). Polar quantizer for wireless receivers: theory, analysis, and cmos implementation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **61**(3), 877–887.

Neuhoff, D. L. (1997). Polar quantization revisited. In *Proc. International Symposium on Information Theory*, page 60. IEEE.

No, A., Ingber, A., and Weissman, T. (2016). Strong successive refinability and rate-distortion-complexity tradeoff. *IEEE Transactions on Information Theory*, **62**(6), 3618–3635.

Pearlman, W. (1979). Polar quantization of a complex gaussian random variable. *IEEE Transactions on Communications*, **27**(6), 892–899.

Pearlman, W. and Gray, R. (1978). Source coding of the discrete fourier transform. *IEEE Transactions on information theory*, **24**(6), 683–692.

Perić, Z. and Nikolić, J. (2013). Design of asymptotically optimal unrestricted polar quantizer for gaussian source. *IEEE Signal Processing Letters*, **20**(10), 980–983.

Peric, Z. H. and Stefanovic, M. C. (2002). Asymptotic analysis of optimal uniform polar quantization. *AEU-International Journal of Electronics and Communications*, **56**(5), 345–347.

Petković, M. D., Perić, Z. H., and Jovanović, A. Ž. (2011). An iterative method for optimal resolution-constrained polar quantizer design. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, **30**(2), 574–589.

Pobloth, H., Vafin, R., and Kleijn, W. B. (2005). Multivariate block polar quantization. *IEEE transactions on communications*, **53**(12), 2043–2053.

Ravelli, E. and Daudet, L. (2007). Embedded polar quantization. *IEEE Signal Processing Letters*, **14**(10), 657–660.

Rimoldi, B. (1994). Successive refinement of information: Characterization of the achievable rates. *IEEE Transactions on Information Theory*, **40**(1), 253–259.

Senge, G. H. (1977). *Quantization of Image Transforms with Minimum Distortion*. Ph.D. thesis, University of Wisconsin–Madison.

Shilov, G. E., Silverman, R. A., *et al.* (1996). *Elementary real and complex analysis*. Courier Corporation.

Skodras, A., Christopoulos, C., and Ebrahimi, T. (2001). The jpeg 2000 still image compression standard. *IEEE Signal processing magazine*, **18**(5), 36–58.

Swaszek, P. (1985). Uniform spherical coordinate quantization of spherically symmetric sources. *IEEE transactions on communications*, **33**(6), 518–521.

Swaszek, P. and Ku, T. (1986). Asymptotic performance of unrestricted polar quantizers (corresp.). *IEEE Transactions on Information Theory*, **32**(2), 330–333.

Swaszek, P. F. and Thomas, J. B. (1982). Optimal circularly symmetric quantizers. *Journal of the Franklin Institute*, **313**(6), 373–384.

Taubman, D. and Marcellin, M. (2012). *JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice*, volume 642. Springer Science & Business Media.

Trushkin, A. (1982). Sufficient conditions for uniqueness of a locally optimal quantizer for a class of convex error weighting functions. *IEEE Transactions on Information Theory*, **28**(2), 187–198.

Vafin, R. and Kleijn, W. B. (2005). Entropy-constrained polar quantization and its application to audio coding. *IEEE transactions on speech and audio processing*, **13**(2), 220–232.

Viswanathan, H. and Berger, T. (2000). Sequential coding of correlated sources. *IEEE Transactions on Information Theory*, **46**(1), 236–246.

Vogel, P. (1995). Source coding by classification-91-715. *IEEE Transactions on Communications*, **43**(11), 2821–2832.

Wallace, G. K. (1992). The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, **38**(1), xviii–xxxiv.

Walrand, J. and Varaiya, P. (1983). Optimal causal coding-decoding problems. *IEEE Transactions on Information Theory*, **29**(6), 814–820.

Wang, C.-Y. and Gastpar, M. (2014). On distributed successive refinement with lossless recovery. In *Proc. International Symposium on Information Theory*, pages 2669–2673. IEEE.

Wiegand, T., Sullivan, G. J., Bjontegaard, G., and Luthra, A. (2003). Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, **13**(7), 560–576.

Wilson, S. (1980). Magnitude/phase quantization of independent gaussian variates. *IEEE Transactions on Communications*, **28**(11), 1924–1929.

Witsenhausen, H. (1979). On the structure of real-time source coders. *Bell System Technical Journal*, **58**(6), 1437–1451.

Wood, R. G., Linder, T., and Yüksel, S. (2017). Optimal zero delay coding of markov sources: stationary and finite memory codes. *IEEE Transactions on Information Theory*, **63**(9), 5968–5980.

Wu, X. (1991). Optimal quantization by matrix searching. *Journal of algorithms*, **12**(4), 663–673.

Wu, X. and Zhang, K. (1993). Quantizer monotonicities and globally optimal scalar quantizer design. *IEEE Transactions on Information Theory*, **39**(3), 1049–1053.

Yuksel, S. (2013). On optimal causal coding of partially observed markov sources in single and multiterminal settings. *IEEE Transactions on Information Theory*, **59**(1), 424–437.