QUANTIZATION DESIGN
FOR STRUCTURED OVERCOMPLETE EXPANSIONS

by

Baltasar Beferull-Lozano

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

December 2002

Dedication

To my outstanding wife, Julia,
an incalculable treasure I found in my life.

To my parents, Vicente and Mª Rosa,
a source of continuous and unconditional love and support.

Acknowledgments

This thesis has been an incredibly exciting, prolific and enjoyable journey which started thanks to my advisor, mentor and friend, Professor Antonio Ortega. I would like to thank him for taking me on board in his group, hence, bringing me to the amazing world of research, and for providing the continuous guidance, enthusiasm and encouragement to make me follow very well the learning curve. His vision, insights and suggestions did not only fuel enormously this research but will have a lasting influence in all my career. You always believed in me and gave me faith all along that we would find that “nugget”, and I think we found a good one. I also want to thank you for the freedom you have always given me in choosing the research topics I most wanted to pursue and enjoy at any time during my thesis, even though some of them were outside the usual research trend in the group. Thanks for trusting me from the first moment.

I would also like to thank all my friends and colleagues at the Signal and Image Processing Institute and at the Communications Science Institute at USC, especially, Wendi Pan, Sangyong Lee, Naveen Srinivasamurthy, Zhourong Miao,

Wenging Jiang, Young-Gap Kwon, Lavanya Vasudevan, Phoom Sagetong, Hua Xie, Raghanvedra Singh, Krisda Lengwehasatit, Guangcai Zhou, Francis and Robert Wilson, for creating always a very good atmosphere, for sharing and discussing all kinds of ideas and for being always there whenever I needed help from them in anything.

I have had also the great pleasure to work with a number of people at AT&T Shannon Laboratory, Florham Park, New Jersey, during my thesis. My work at AT&T Shannon Laboratory as a Ph.D. intern not only has led to several joint publications and an important part of my thesis, but has influenced me importantly in many positive ways. I thank Zoran Cvetković and Vinay Vaishampayan for organizing my trip to Florham Park and also for all the fruitful discussions and collaborations in the different research projects I was involved in. I learned many things from both of them. I would also like to thank Neil Sloane, Suhas Diggavi, Amy Reibman, Andreas Buja, Bishnu Atal, Jont Allen, Anna Gilbert, Sarangarajan Parthasarathy, Robert Calderbank and Ingrid Daubechies, for all the discussions and support throughout my stay at AT&T Shannon Laboratory. In particular, I really want to thank Neil Sloane, "the Integer man", for getting so interested and excited about my research and for working with me with an intensity that I had never seen before in my life. I will never forget the incredible experience of working with him for many reasons. First, the permanent enthusiasm and creativeness he has is extremely contagious and makes you enjoy every moment of research work with him. Second, even though he is a complete human encyclopedia, he was always ready to listen or learn any possible thing from my comments or thoughts, treating me with a great respect. This willingness to keep learning as much as possible from anyone is, in my opinion, one clear indicator of the fact that he is a real genius. Third, during my work with him, I learned an incalculable amount of useful and beautiful things about lattice theory that he taught me with incredible patience. As I said to him so many times: Thanks Neil!. I also thank Suhas Diggavi, who after I left from AT&T Shannon Laboratory, has been continuously collaborating with me.

I also want to take this opportunity to thank my Professors at USC, from which I have learned so many useful things in the different courses. The background I received from many of the courses was very useful for me during my research work. I thank all of them for their dedication to teaching and for being always so open for discussion. In particular, I would like to thank Professor Vijay Kumar, for being always very supportive and for taking the time to discuss with me about my research work.

Prof. Robert Guralnick, Prof. Won Namgoong, Zoran Cvetković, I thank you for serving on my Qualifying exam committee. Prof. Robert Penner, I thank you for serving on my Defense committee. Prof. Vijay Kumar, I thank you for serving on my Qualyfing exam and Defense committees.

My deepest and most sincere gratitude goes to my outstanding wife Julia. I am sure I would not have achieved this long-sought goal without her. I would need many many pages to say all the things I have to thank my wife for, so I would have to summarize it here very briefly. First, I want to really thank my wife for giving me so much support, love and happiness, without which, I could not have completed this thesis. Second, I want to thank my wife for sacrificing so much time of her life for me so that I could work intensively on my research during all these years. Finally, I want to thank her for taking care of me so well and being a source of continuous positiveness in life that still surprises me every day. Gracias Julia!

I want to thank my parents, Vicente and Mª Rosa for all their unconditional help and support. My mom has given me all the loving care that only a mother can provide. My dad has always supported me in all of my endeavors, trusting me from the very first moment. He has shown me through all his life the importance and the power of the work well done and I have followed him as a role model. My father and mother in-law, Miguel and Mª Consuelo, I thank them for being always ready to help me and my wife whenever it was needed and for showing me through their example the importance of looking always at the positive side of life, one thing that my wife and her daughter Julia has learned so perfectly.

I also want to thank Prof. Francisco Toledo Lobo from Universitat Jaume I, Castellón, Spain, and Prof. Gregorio Martín, from Universidad de Valencia,

Valencia, Spain, who gave me a very valuable initial support in my different endeavors while I had a joint research assistant position both in the Department of Electronics and Computer Science at Universidad de Valencia and in the Department of Computer Science at Universitat Jaume I.

Last, but certainly not least, I gratefully acknowledge the generous support of the Ministry of Science and Education from Spain through a Doctoral Fellowship, National Science Foundation (NSF), NASA and AT&T Shannon Laboratory, which made possible this research during my Ph.D.

Contents

[1] The publications related to this chapter are [11, 10, 8].

[2] Some of the work in this chapter was published in [102]. This work was carried out in collaboration with N. J. A. Sloane at AT&T Shannon Laboratory.

[3] The publications related to this chapter are [9, 88, 13, 14].

[4] This is current work being carried out in collaboration with Suhas Diggavi from AT&T Shannon Laboratory. Part of this work is to be published in[12].

List of Tables

List of Figures

Abstract

The study of quantized overcomplete (redundant) expansions is relevant to several important applications such as oversampled A/D conversion of band-limited signals, multiple description quantization, joint source-channel coding and content-based retrieval for image databases. The problem of quantization of overcomplete expansions is not as well understood as that of quantization of more traditional critically sampled (non-redundant) expansions. As an example, in the latter case, one can minimize the overall distortion by minimizing the distortion independently in each of the expansion coefficients. This is not true for an overcomplete expansion.

Previous research work to date has assumed only simple quantization schemes and has focused on finding improved reconstruction algorithms. In our work, we study different issues related to overcomplete expansions, focusing on designing efficient quantization techniques for this type of decompositions. More specifically, the following topics are studied:

(i) We propose new quantization designs for overcomplete expansions in $\mathbb{R}^N$. Our approach is to design jointly the overcomplete decomposition together with the quantization scheme so that the whole system is equivalent to a regular vector quantizer in $\mathbb{R}^N$ having a periodic structure. The periodic structure can be characterized by using lattice intersections.

(ii) We show how the periodicity property makes it possible to achieve good accuracy with low complexity, by analyzing linear reconstruction in periodic quantizers and providing also some other low complexity reconstruction schemes which can be implemented thanks to the presence of a periodic structure.

(iii) Given an intersection lattice Λ, we provide general methods to decompose it as the intersection of simpler lattices which are nested in Λ. We also give concrete decompositions for most of the best known lattices giving rise to different periodic quantizers with different tesselations.

(iv) We obtain an expression for the effective normalized second moment G of a periodic quantizer, which characterizes its rate-distortion performance at high rates. We analyze the complete structure of the tesselations generated by some of the derived lattice decompositions and evaluate the value of G for the corresponding periodic quantizers.

(v) We analyze angular oversampling in the presence of quantization for overcomplete $2D$ filter banks in $\ell^2(\mathbb{Z}^2)$ which are steerable under rotation. We define two "consistency" constraints, one due to the steerability property and the other one due to the quantization itself, and make use of them in order to increase the accuracy in the representation with the number of orientations by using two main techniques in conjunction with Lie theory: a) Projection on Convex Sets (POCS) and b) Linear programming principles.

(vi) We define energy-based features which are steerable under rotation and which are based on steerable transforms, and apply them to the problem of Rotation Invariance in content-based image retrieval. We define a similarity measurement where the features of different images are compared only after they have been previously aligned. This capability can not be achieved by a regular wavelet transform.

Chapter 1

Introduction

Quantized redundant expansions are useful in different applications such as oversampled A/D conversion of band-limited signals [39, 40, 42, 84, 111, 60], multiple description quantization [59][58], joint source-channel coding [53], content-based retrieval [88] and pattern recognition [101, 99]. The overcompleteness provides two important advantages, namely, an increase in design freedom [38] and robustness against noise. One of the most important applications is the analog-to-digital conversion of band-limited signals, where instead of achieving accuracy by using high rate quantization of the amplitude (having to use complex and expensive analog circuitry of high precision), accuracy is attained by performing an oversampling in the time axis (timing accuracy is easier to achieve in VLSI), that is, sampling at a rate above the Nyquist rate, and exploiting this redundancy to reduce the information which is lost after quantizing the amplitude with low resolution (oversampled A/D).

The accuracy that can be attained with quantized overcomplete expansions depends on two factors: the reconstruction algorithm and the quantization scheme that are used. In the context of overcomplete expansions in $\mathbb{R}^N$, as we will see in Chapter 2 an equivalent vector quantizer (EVQ) can be defined given a quantized redundant expansion, where the quantized vector is given by the reconstruction vector obtained from the quantized coefficients of the redundant expansion by applying some reconstruction algorithm. That is, projecting a signal $\boldsymbol{x}$ on an overcomplete set of vectors, applying a set of scalar quantizers over the corresponding set of coefficients and obtaining a reconstructed vector $\hat{\boldsymbol{x}}$ from the quantized coefficients using some reconstruction algorithm can be seen as equivalent to a vector quantizer EVQ in $\mathbb{R}^N$ where $EVQ(\boldsymbol{x}) = \hat{\boldsymbol{x}}$.

There has been extensive research work aimed at finding reconstruction algorithms that maximize the asymptotic rate of decrease of MSE with the redundancy of the overcomplete expansion. Reconstruction algorithms have been studied based on two main approaches. The first one is based on modeling the quantization noise as an additive white noise uncorrelated with the signal that is quantized. These models are sometimes convenient for analysis and lead to useful results in some scenarios [16, 63]. The second one is completely based on a deterministic analysis of the quantization noise, which have given rise to several proposed reconstruction algorithms [84, 111, 60, 40]. We discuss in Chapter 2 the different issues related to these algorithms and the simple linear reconstruction algorithm.

However, the quantization scheme has been always assumed to be a uniform scalar quantization with the same stepsize for all expansion coefficients. One of the issues that is addressed in our work is to study whether using different stepsizes to quantize the coefficients can lead to improvements in the rate-distortion performance when reconstructing with simple reconstruction algorithms. Notice that simple reconstructions (e.g. linear or look-up table) are normally prefered for practical reasons. In general, our goal is to have an encoder having a complexity which is similar to that of standard systems, allowing to use simple reconstruction algorithms without a loss in accuracy with respect to more sophisticated reconstruction techniques. The fundamental idea to achieve this goal is to design jointly the overcomplete expansion together with the quantization system by choosing carefully the stepsizes of the scalar quantizers so that the whole system is equivalent to a vector quantizer EVQ in $\mathbb{R}^N$ with a periodic structure. This periodic structure can be conveniently characterized and parameterized in terms of lattice intersections, more specifically, the overcomplete expansion together with the set of scalar quantizers effectively generate a set of different lattices $\{\Lambda^1, \Lambda^2, \cdots, \Lambda^r\}$ such that their intersection Λ is not empty. The periodicity of the quantizers has a unit cell whose structure is repeated and this unit cell is given by the fundamental Voronoi cell of the intersection lattice that we call

Coincidence site lattice. We show that periodicity in the EVQ is a necessary condition in order for the EVQ to be a regular quantizer.

From this connection with lattice theory, in Chapter 3, we then consider the construction of periodic quantizers such that the intersection lattice is a good lattice. Thus, given a good lattice Λ (e.g., such that Q_Λ is a good lattice quantizer), we want to find a set of simple lattices $\{\Lambda^1, \Lambda^2, \cdots, \Lambda^r\}$ such that their intersection is not empty. The main reason for investigating this problem is that if Λ is a good lattice, we expect the multiple description quantizer $(Q_{\Lambda^1}, \cdots, Q_{\Lambda^r})$ which quantizes simultaneously an input vector $\boldsymbol{x}$ with respect to each of the Λ^i's, to be also a good periodic quantizer. We analyze general constructions for writing a lattice as an intersection and more specifically, our focus is in the following question: which good lattices in different dimensions have a description as the intersection of lattices such that a) the lattices in the decomposition are as simple as possible (ideally cubic lattices) and b) the symmetry properties are as good as possible? We give some answers and provide concrete decompositions for most of the best known lattices, that is, for the lattices whose corresponding lattice quantizers are the best quantizers currently known, assuming high-rate quantization. Then, we characterize the rate-distortion performance of these periodic quantizers by deriving an expression for the dimensionless normalized second moment, which is a generalization of the normalized second moment of a lattice quantizer. We calculate the corresponding tesselations for some of the decompositions we derive and evaluate the rate-distortion performance of the associated periodic quantizers.

In the context of $\ell^2(\mathbb{Z})$, we study signal representation of oversampled steerable transforms in the presence of quantization. We focus our study on transforms which are steerable under rotation although we also introduce some general theory which is valid for any transformation Lie group. The angular oversampling in the context of steerable transforms has not been considered by prior research and we explore techniques to represent efficiently this oversampled data. The angular oversampling or oversteering is also motivated because it allows us to establish some "consistency" constraints [39, 60, 84, 111] on the coefficients of a steerable representation with many orientations (oversteered representation) which reduces the amount of information lost in the quantization process and thus increases the accuracy and resolution of the corresponding coefficients. Two methods are given in order to impose these consistency constraints, one based on projection on convex sets (POCS) where the convex sets are the the sets of (steerability and quantization) constraints, and the other one based on linear programming by calculating regions of uncertainty in the angular domain.

Then, we study the problem of Rotation Invariance in the context of a content-based image retrieval application. We define a set of energy-based features which are actually steerable, that is, given the features of a texture oriented at an angle ϕ, the features corresponding to the same texture but oriented at an angle ϕ', can be approximately estimated from the features at angle ϕ without actually having to calculate the features for the rotated version. This is a very useful property because it allows to use a similarity measurement such that the features of two different images are aligned before they are compared. In a steerable pyramid with more than one scale or level, all the features across all the levels are steered at the same time in order to be consistent with what would happen if the original image was rotated. Notice that this alignment can not be done with a regular wavelet transform because of the problem of rotation-variance in these critically sampled transforms.

The main contributions described in this thesis include:

- Efficient Quantization for overcomplete expansions in $\mathbb{R}^N$:

    1. Joint design of quantization systems and structured overcomplete expansions in $\mathbb{R}^N$ which allow to use simple reconstruction algorithms with low complexity while having good performance in terms of accuracy.

    2. Complete formulation of quantizers with periodic structure in terms of lattice theory, arising the concept of lattice intersections and geometrically scaled-similar sublattices. Necessary and sufficient conditions for achieving a (purely geometrical) periodic structure are given.

3. The quantized overcomplete expansion induces an equivalent vector quantizer (EVQ). Necessary conditions in order to have an EVQ which is regular under linear reconstruction are given.

4. The periodicity is what makes it possible to achieve good accuracy using simple reconstruction algorithms. We show this by providing some low complexity reconstruction schemes which can be used thanks to the periodicity in the structure. Experimental results show the superiority of our scheme.

5. Implications of our work in (simple encoding) oversampled A/D conversion of sinusoid signals are also shown.

- Periodic Quantizers based on good Lattice Intersections; Construction and Analysis:

  1. We describe different general methods in order to write a lattice as an intersection of several simpler lattices such that the symmetry properties are maximized.

  2. We provide decompositions for most of the best known lattices (the hexagonal lattice A_2, the face-centered lattice D_3, the body-centered cubic D_3^* lattice, the root lattices D_4, E_6^*, E_8, the Coxeter-Todd, Barnes-Wall and Leech lattices, etc.) in a canonical way as intersections of a small number of simpler, decomposable, lattices.

3. Assuming that Λ is the intersection of lattices $\Lambda^1, \ldots, \Lambda^r$, we analyze the tesselation that is obtained as a consequence of simultaneously quantizing $\boldsymbol{x}$ with respect to each of the Λ^i. The cells of this tesselation are the intersections of the Voronoi cells for the Λ^i and the output of the quantizer is given by the barycenter of the cell to which $\boldsymbol{x}$ belongs. We analyze in depth the geometry of the tesselations for the cases where the intersection lattices are A_2, D_3, D_3^* and D_4, which have not been studied in previous research.

4. A generalization of the expression for the (dimensionless) normalized mean squared error G is obtained for the case of a quantizer with a periodic tesselation, obtaining in this way a figure of merit for these periodic quantizers. Our formulation includes also the case of lattice vector quantizers as a particular case.

- Quantization in oversampled steerable transforms:

    1. Energy compaction in angle (as we increase the number of orientations in the steerable transform) is obtained by performing simple thresholding on the (angles) coefficients of maximum energy for each location. This gives rise to a simple method to represent the oversampled data based on a selection of maximums (in energy) across the different orientations.

2. Use of projection on convex sets (POCS) to improve accuracy with two convex constraints: a) angular consistency constraints due to the steerability property and b) the constraints induced by the quantization itself. Experimental results are obtained showing a coding gain for low rates as we increase the number of orientations (for a certain range of orientations).

3. A general formulation based on linear programming is obtained to calculate regions of uncertainty in the angular domain. Different properties and theoretical results are given showing that the uncertainty about the transform coefficients is reduced as we increase the number of orientations.

4. A new Rotation-Invariant content-based image retrieval system is proposed based on steerable pyramids. We define energy-based features that are steerable and allow to use a similarity measurement between textures such that it performs an alignment of features, which is a property that can not be achieved with a regular wavelet transform. We derive the equation describing the steering of the features and show how to use it not only to measure similarity between textures but also in order to estimate the relative angles between any two rotated versions of the same texture. Experimental results testing the rotation

invariance show a clear advantage of using steerable transforms instead of regular wavelets.

Finally, current and future work is described in some detail in Chapter 5, with an emphasis on topics (e.g. power shaping for the Costa problem) that are the object of current work.

Chapter 2

Efficient Quantization for Overcomplete Expansions in $\mathbb{R}^N$*

2.1 Introduction and Motivation

Quantized redundant expansions are useful in different applications such as oversampled A/D conversion of band-limited signals [39, 40, 42, 84, 111, 60] and multiple description quantization [59, 58, 57, 26]. In the first case, the purpose of using redundant expansions is to attain accurate digital signal representations under scenarios where the cost of using high rate quantization is much higher than that of having a high oversampling or redundancy. The most important case is the analog-to-digital conversion of band-limited signals, where in order to use high rate quantization to discretize the amplitude it is necessary to use expensive

*The publications related to this chapter are [11, 10, 8].

high precision analog circuitry. Instead, accuracy is attained by performing oversampling and exploiting this redundancy to reduce the loss of information caused by low resolution quantization. A more sophisticated strategy is used in sigma-delta based converters, where the fact that the signal is band-limited is further used by performing prediction between samples and reducing more effectively the energy of noise which has more correlation with the input signal, allowing the use of very simple quantizers (typically single-bit quantizers). Some other systems have been proposed in the context of pattern recognition for images, where overcomplete transforms are used to emulate the human visual system, which has a high degree of oversampling in orientation and scale [70, 71]. More specifically, the visual (striate) cortex is organized in columns containing simple cells, which act as oriented two dimensional linear filters, so that the cells in each column have receptive fields (playing the same role as filter impulse responses) which have roughly the same orientation (although varying in size or scale by octaves) and the orientations of the cells belonging to adjacent columns differ by about only ten degrees [70, 71]. The effective number of orientations and scales that are present is clearly much higher than what is needed to represent the visual signal comming into the visual system. An increase in resolution (reduction of quantization of error) due to angular oversampling in the frequency domain has been observed experimentally for quantized (two-dimensional) steerable transforms, so that increasing the number of orientations yields a gain in energy compaction

[9]. Quantized overcomplete expansions also arise in the context of joint source-channel coding for erasure channels [59, 58, 57, 26].

There are two major factors that determine the accuracy that can be attained using quantized overcomplete expansions: the reconstruction algorithm and the quantization scheme. There has been extensive research work aiming at finding reconstruction algorithms that are optimal or near optimal in terms of asymptotic (large redundancy values) accuracy. However, the quantization scheme has been always assumed to be a uniform scalar quantization with the same stepsize for all expansion coefficients. In this work, we explore efficient quantization designs for overcomplete expansions.

Reconstruction algorithms have been studied following two main approaches. The first one is based on modeling the quantization noise as an additive white noise uncorrelated with the signal that is quantized. These models are sometimes convenient for analysis and lead to useful results in some scenarios [16, 63]. It can be shown that if a white noise model is assumed for the scalar quantization noise of the coefficients and the same stepsize is used to quantize all the coefficients, the optimal reconstruction is given by the usual linear reconstruction [44], where linear reconstruction consists of first projecting the signal into a set of vectors (with cardinality larger than the dimension) obtaining a set of coefficients, and then reconstructing by taking a simple weighted average of these coefficients. Thus, in practice, linear reconstruction is always used when the assumptions leading to this analysis are valid. In the context of tight frames, an important class of overcomplete expansions, theoretical analysis shows (under this quantization scheme and stochastic model) that linear reconstruction [44] gives a reduction in the power of each noise component (quantization noise of each projection or coefficient) that is proportional to the redundancy r of the tight frame. The same decay of the MSE in the signal domain can be shown theoretically in the cases of tight frames in $\mathbb{R}^N$, Weyl-Heisenberg frames in $\ell^2(\mathbb{Z})$ and in classical oversampled A/D conversion with uniform sampling and linear reconstruction (tight sinc frames) where $MSE = \frac{\Delta^2}{12r}$ [16]. The behavior of the $MSE = O(1/r)$ is observed experimentally when uniform quantization with the same stepsize is used and the stepsize is small enough so that the white noise model approximately applies. One of the reasons for linear reconstruction not to be optimal in some cases is that the reconstructed signal may not be *consistent* with the original signal in the sense that the output obtained from requantizing the reconstructed signal is different than the output obtained when quantizing the original signal, implying a larger reconstruction error on average. On the other hand, it has not been studied whether using a more intelligent quantization system allowing in general different stepsizes to quantize the coefficients can lead to improvements in the rate-distortion performance when reconstructing with a linear reconstruction algorithm. This is one of the issues that is addressed in this work.

The second approach is completely based on a deterministic analysis of the quantization noise. This deterministic approach was introduced by Thao and Vetterli [84, 111] and later extended in [39, 42, 60]. This deterministic analysis based on hard bounds of the quantization noise led to two non-linear reconstruction algorithms for frames in $\mathbb{R}^N$, one based on projection on convex sets theory (POCS) [130, 84, 111, 66, 65] and the other one based on linear programming (LP) [60]. The main result is that these reconstruction algorithms ensure that the reconstruction vector falls always inside the same cell as the input vector. These reconstructions are called *consistent* and in quantization terms this means that the equivalent quantizer is regular. It was observed experimentally that for high enough redundancies r and for uniform quantization of all the frame coefficients consistent reconstruction algorithms have an asymptotic MSE behavior of $O(1/r^2)$. Moreover, Thao and Vetterli proved (under some mild conditions) that consistency guarantees this asymptotic behavior for high enough redundancies r for the case of oversampled A/D conversion of T-periodic bandlimited continuous-time signals, which can be viewed as a frame expansion in $\mathbb{R}^N$ with respect to a certain DFT-like frame. Later, Cvetković [39, 40] proved this fact under some mild restrictions for overcomplete expansions in $\mathbb{R}^N$ in general. Cvetković proposed a more efficient reconstruction algorithm called *semilinear reconstruction* *algorithm* which also attains asymptotically an accuracy of $O(1/r^2)$ without satisfying consistency. This algorithm is based on the positions of the threshold crossings and identifying a good linear system to solve. Moreover, Cvetković and Daubechies extended this idea to be used in the context of single-bit oversampled A/D conversion where a deterministic dither is used in order to force threshold crossing locations with certain properties which allow exponential accuracy in the bit-rate [41]. Recently, Rangan and Goyal [92] have proposed a recursive algorithm using subtractive dithered quantization which also attains asymptotically an accuracy of $O(1/r^2)$, again, without ensuring consistency.

The crucial observation that motivates our work is that in all the previous work a very simple quantization scheme has been assumed which requires sophisticated reconstruction algorithms [130, 84, 111, 66, 65, 92, 39, 40] in order to improve its accuracy with respect to the classic approach [44] (simple quantization and linear reconstruction). Instead, we pose the following question: are there quantization schemes where there is no difference in performance between using simple reconstruction algorithms (e.g., linear or of similar complexity) and more sophisticated reconstruction methods? Although all the improved reconstruction algorithms that have been proposed so far can achieve very good accuracy, the computational complexity of these methods (although different in each case), for a given redundancy, is higher than that of linear reconstruction [60, 84, 111]. Since simple reconstructions (e.g., linear or look-up table) are normally preferable in practical scenarios, in our work we assume that a simple reconstruction will be used and the main focus is to explore whether better quantization designs, e.g., using different stepsizes, may have the advantage of achieving a performance which is superior with respect to simple quantization methods, e.g., using the same stepsize. In other words, our goal is to provide the tools to design the overcomplete expansions and the corresponding quantization system so that the overall system behaves like a regular quantizer and achieves the best possible performance using simple reconstruction algorithms. Designing the quantization system with a structure that forces consistency, using the usual linear reconstruction, may result in worse performance, in terms of rate distortion, than a *different* system whose structure results in inconsistency. However, we will show that because of the periodic structure of the quantization system, very simple reconstruction techniques (e.g., those based on a look-up table) can be designed which significantly outperform linear reconstruction.

The fundamental idea we use in order to achieve this goal is to design jointly the overcomplete expansion together with the quantization system by choosing carefully the stepsizes of the scalar quantizers so that the whole system is equivalent to a vector quantizer in $\mathbb{R}^N$ with a periodic structure. First, we define an equivalent vector quantizer given a quantization scheme and a reconstruction algorithm. Then, based on this equivalence, we introduce the concept of periodic quantizers and show how to construct and design periodic quantizers. This periodic structure can be conveniently characterized and parameterized in terms of lattices and sublattices. The construction we give in order to achieve periodicity is completely general, while we provide designs for specific cases of redundant families of vectors. More specifically, in this chapter, almost all the designs are given for $\mathbb{R}^2$ and where the redundant families have a certain constrained structure. Designs for higher dimensions are provided in Chapter 3. Next, we explain the advantages that are provided by this periodic structure and show how the periodic structure in the equivalent vector quantizer is a necessary condition to achieve consistency under the usual linear reconstruction. This result holds for any arbitrary frame where the reconstruction is linear. Once a periodic structure is present, the number of different cells of this vector quantizer becomes finite and although a sufficient condition can not be expressed formally, it is very simple to check whether consistency is satisfied or not. For a given family of vectors and a set of different stepsizes which yield a periodic vector quantizer in $\mathbb{R}^N$, it is possible to reconstruct by using a small look-up table, where the reconstruction vectors can be chosen to be the centroids of the cells with respect to a uniform distribution. Moreover, it is also possible to design systems such that the equivalent vector quantizer has some additional symmetry which allows to use a very simple improved linear reconstruction. Our system provides excellent performance while having the same complexity as linear reconstruction, but is more suitable to be used in $\mathbb{R}^N$ for low to moderate values of the dimension N and for low values of redundancy [10, 11]. Although we present examples and results for small redundancies, it is clearly shown that the basic theoretical idea of periodicity can be extended to higher redundancies and that the problem of finding good quantizers with higher redundancies consists of searching for good lattices and sublattices with certain properties. Extensions to higher dimensions have been analyzed by Sloane and Beferull-Lozano recently and can be found in [102] and Chapter 3. On the other hand, although we believe that multiple description coding is also a potential application of our framework, we have not explored this application in this work.

This chapter is organized as follows. In section 2.2 we define the equivalent vector quantizer and the property of consistency. Section 2.3 describes the construction and design of periodic quantizers in terms of lattices. In section 2.4, it is first shown that the periodic structure in the equivalent vector quantizer is a necessary condition to achieve consistency under the usual linear reconstruction, and then, low complexity reconstruction schemes in periodic quantizers are analyzed. Finally, numerical results for some specific designs in $\mathbb{R}^2$ are shown in section 2.5 as well as a simple direct application of our designs in $\mathbb{R}^2$ to oversampled A/D conversion of sinusoid signals.

2.2 Linear Reconstruction, Equivalent VQ and Consistency

In this section, we first review the basic concept of a tight frame in $\mathbb{R}^N$, and express a linear reconstruction in terms of an equivalent vector quantizer (EVQ), which can be parameterized in terms of lattices.

2.2.1 Linear reconstruction in tight frames without quantization

For the sake of clarity, we review briefly the definitions and main properties of tight frames.

Definition 1 *Let* $\Phi = \{\boldsymbol{\varphi}_i\}_{i=1}^{M} \subset \mathbb{R}^N$ *where* $\|\boldsymbol{\varphi}_i\| = 1$, $\forall\ i = 1, \dots, M$. Φ *is called a frame if there exist* $A > 0$ *and* $A \leq B < \infty$ *such that*

$$A\|\boldsymbol{x}\|^2 \leq \sum_{i=1}^{M} |\langle \boldsymbol{x}, \boldsymbol{\varphi}_i \rangle|^2 \leq B\|\boldsymbol{x}\|^2, \qquad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{2.1}$$

A and B are called lower and upper frame bounds. Given a frame Φ, the associated frame operator $\boldsymbol{F} : \mathbb{R}^N \longrightarrow \mathbb{R}^M$ is given by an $M \times N$ matrix defined as:

$$\begin{aligned} \boldsymbol{F} &= (\boldsymbol{\varphi}_1 \boldsymbol{\varphi}_2 \dots \boldsymbol{\varphi}_M)^T \\ \boldsymbol{y}_i &= (F\boldsymbol{x})_i = \langle \boldsymbol{x}, \boldsymbol{\varphi}_i \rangle = \boldsymbol{\varphi}_i^T \boldsymbol{x} \qquad \forall \boldsymbol{x} \in \mathbb{R}^N \end{aligned} \tag{2.2}$$

Definition 2 *The minimal dual frame of* Φ *is defined as* $\widetilde{\Phi} = \{\widetilde{\boldsymbol{\varphi}}_{\boldsymbol{i}}\}_{i=1}^{M}$ *where:*

$$\widetilde{\boldsymbol{\varphi}}_i = (\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{\varphi}_i \qquad \forall i = 1, \ldots, M. \tag{2.3}$$

Definition 3 *A frame* Φ *is called a tight frame if* $A = B$*, that is, if the lower and upper bounds are equal.*

The following properties are satisfied for a tight frame:

1. The minimal dual frame $\widetilde{\Phi}$ of a tight frame Φ is given by:

$$\widetilde{\boldsymbol{\varphi}}_{\boldsymbol{i}} = \frac{1}{r}\boldsymbol{\varphi}_i \qquad \forall i = 1, \ldots, M \qquad \text{with } r = \frac{M}{N} \tag{2.4}$$

and the redundancy r of the tight frame is equal to the frame bounds, that is, $r = A = B$.

2. $\forall\ \boldsymbol{x} \in \mathbb{R}^N$, the expansion with respect to the frame $\boldsymbol{\Phi} = \{\boldsymbol{\varphi}_i\}_{i=1}^{M}$ whose coefficients have the minimum possible norm (most economical expansion), is given by:

$$\boldsymbol{x} = \sum_{i=1}^{M}\langle \boldsymbol{x}, \widetilde{\boldsymbol{\varphi}}_i\rangle\boldsymbol{\varphi}_i = \frac{1}{r}\sum_{i=1}^{M}\langle \boldsymbol{x}, \boldsymbol{\varphi}_i\rangle\boldsymbol{\varphi}_i. \tag{2.5}$$

In this section we restrict the discussion to the case of tight frames that are composed by a set of $r > 1$ different orthogonal bases. This is done without loss of generality for purposes of clarity because the geometric analysis is much simpler. Extensions to generic frames are simple and can be obtained by using in the reconstruction the corresponding dual frames, which will be different in each case. With this restriction, we can group the vectors $\{\boldsymbol{\varphi}_i\}_{i=1}^{M}$ that compose the tight frame as $\{\{\boldsymbol{\varphi}_i^j\}_{i=1}^{N}\}_{j=1}^{r}$, where $\{\boldsymbol{\varphi}_i^j\}_{i=1}^{N}$ is the j-th basis.

Remark on Notation: In this chapter, we make an extensive use of superscripts and subscripts. For instance, in a tight frame composed of r orthogonal bases, the superscript $j \in \mathbb{Z}_+$ indicates the j-th basis and the subscript i indicates the i-th vector of the j-th basis. Also, in order to avoid confusion with the superscripts, to represent a number b raised to the power of e (e being any real number), we will use $(b)^e$, and we will use b^e for indexation (e-th element), with $e \in \mathbb{Z}_+$.

For the sake of simplicity, we restrict most of the equations and expressions of this section, without any loss of generality, to the case of $\mathbb{R}^2$. For $N = 2$, the frame contains $M = 2r$ unitary vectors that form r orthogonal bases and the frame operator can be written as $\boldsymbol{F} = [\boldsymbol{\varphi}_1^1 \boldsymbol{\varphi}_2^1 \boldsymbol{\varphi}_1^2 \boldsymbol{\varphi}_2^2 \ldots \boldsymbol{\varphi}_1^r \boldsymbol{\varphi}_2^r]^T$. If we define each orthogonal matrix $\boldsymbol{F}^j$ as $\boldsymbol{F}^j = [\boldsymbol{\varphi}_1^j \boldsymbol{\varphi}_2^j]^T$, then we call $\boldsymbol{y}^j = [y_1^j, y_2^j]^T$ the 2-dimensional vector of coefficients associated with the j-th basis, which is given by $\boldsymbol{y}^j = \boldsymbol{F}^j \boldsymbol{x}$. The M-dimensional vector of coefficients $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$ will be expressed as $\boldsymbol{y} = [y_1^1, y_2^1, y_1^2, y_2^2, \ldots, y_1^r, y_2^r]^T$.

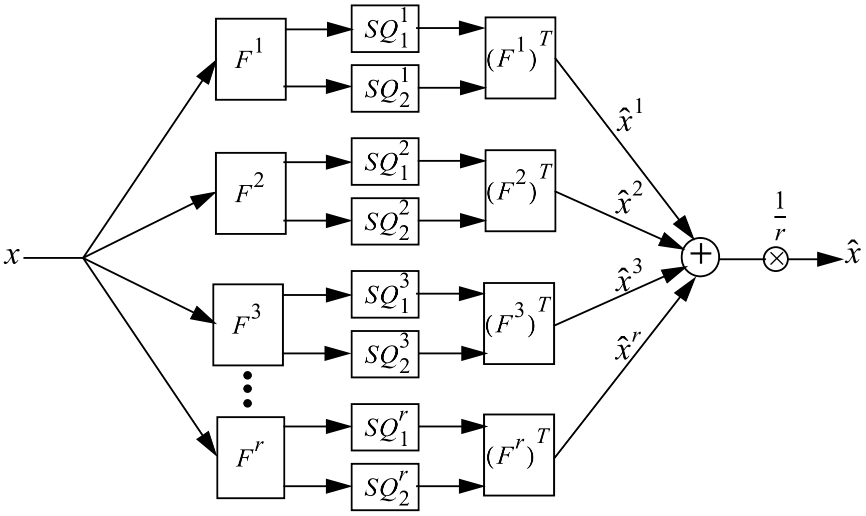
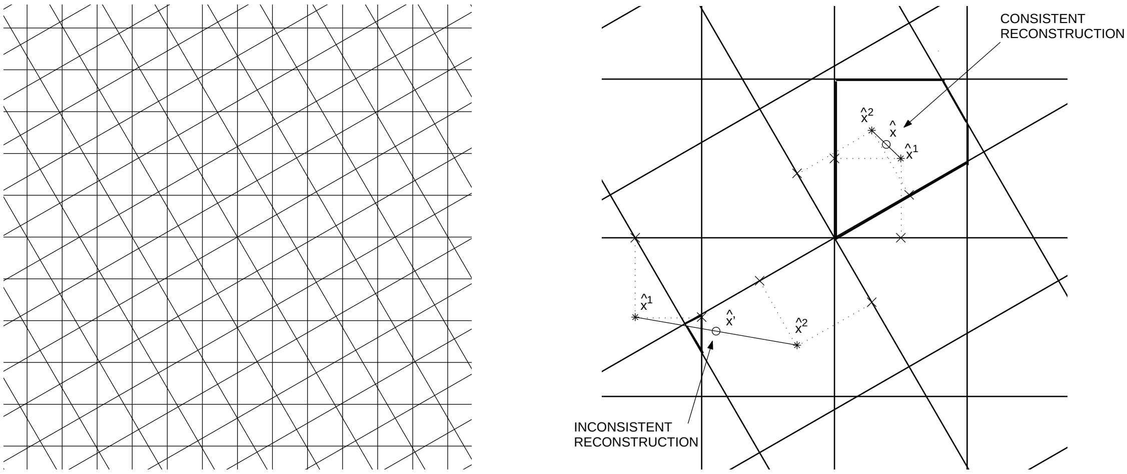


Figure 2.1: Definition of the EVQ in $\mathbb{R}^2$ for a tight frame based on the linear reconstruction given by the minimal dual frame. A similar definition for the EVQ can be given for any general linear reconstruction algorithm.

2.2.2 Equivalent Vector Quantizer (EVQ) for linear reconstruction

Assume that scalar quantization is applied to the frame coefficients. Let SQ_i^j be a uniform scalar quantizer with stepsize Δ_i^j and decision points $\{m\Delta_i^j\}_{m\in\mathbb{Z}}$. This is a particular choice without any loss of generality, that is, what follows is also valid for scalar quantizers with decision points $\{(m+\frac{1}{2})\Delta_i^j\}_{m\in\mathbb{Z}}$ where 0 is a reconstruction point.

Then, we define $SQ_1^1 \times SQ_2^1 \times \ldots \times SQ_1^r \times SQ_2^r$ as an M-dimensional product scalar quantizer (PSQ) applied to the M-dimensional vector of coefficients $\boldsymbol{y}$,



(a) (b)

Figure 2.2: a) Example of the convex polytopes C^{EVQ} in $\mathbb{R}^2$, b) (Zoom) Example of outputs for the quantizers Q^1, Q^2 and the EVQ when linear reconstruction is used. The partial reconstructions $\hat{\boldsymbol{x}}^j$, $j = 1, 2$ are represented by $'*'$ and the final reconstruction $\hat{\boldsymbol{x}}$ is represented by $'\circ'$. The final reconstructions are obtained by taking the halfway point between $\hat{\boldsymbol{x}}^1$ and $\hat{\boldsymbol{x}}^2$, that is, $\hat{\boldsymbol{x}} = \frac{1}{2}(\hat{\boldsymbol{x}}^1 + \hat{\boldsymbol{x}}^2)$. Two reconstructions are shown, each reconstruction corresponding to the case where the original vector $\boldsymbol{x}$ is in each of the two EVQ cells indicated with the bold line. $\hat{\boldsymbol{x}}$ is a consistent reconstruction and $\hat{\boldsymbol{x}}'$ is an inconsistent reconstruction.

i.e., each of the components of the vector $\boldsymbol{y}$ are quantized by a corresponding scalar quantizer (see Fig. 2.1).

Given a tight frame Φ and a PSQ, we define the following quantizer:

Definition 4 *A quantizer Q^j, $1 \leq j \leq r$ consists of:*

1. *A set C^j of rectangular quantization cells induced by the scalar uniform quantizers $\{SQ_1^j, SQ_2^j\}$ which are applied to the frame coefficients associated with the j-th basis.*

2. *A mapping* $\mathbb{R}^2 \longrightarrow \mathbb{R}^2$ *from the set of cells* C^j *to a set of reconstructions (outputs)* O^j *such that* $\forall \boldsymbol{x}$ *satisfying* $m_i \Delta_i^j \leq SQ_i^j(\langle \boldsymbol{x}, \boldsymbol{\varphi}_i^j \rangle) \leq (m_i + 1)\Delta_i^j$, $i = 1, 2$ *the reconstruction vector is given by*

$$\begin{gathered} \hat{\boldsymbol{x}}^j = Q^j(\boldsymbol{x}) = \textstyle\sum_{i=1}^2 SQ_i^j(\langle \boldsymbol{x}, \boldsymbol{\varphi}_i^j \rangle)\boldsymbol{\varphi}_i^j, \qquad SQ_i^j(\beta) = \left(\left\lfloor \frac{\beta}{\Delta_i^j} \right\rfloor + \frac{1}{2} \right) \Delta_i^j \\ \Longrightarrow \hat{\boldsymbol{x}}^j = \textstyle\sum_{i=1}^2 \left(m_i + \frac{1}{2} \right) \Delta_i^j \boldsymbol{\varphi}_i^j. \end{gathered} \tag{2.6}$$

The stepsize associated with the scalar quantizer SQ_i^j is denoted by Δ_i^j. The vertices of the cells C^j form what is called a real 2-dimensional lattice.

Definition 5 *A N-dimensional lattice* Λ *is a discrete subgroup of* $\mathbb{R}^N$ *which is defined as the set of points obtained by taking linear combinations of N linearly independent vectors with coefficients being integers:*

$$\Lambda = \{\boldsymbol{x} : \boldsymbol{x} = u_1\boldsymbol{a}_1 + u_2\boldsymbol{a}_2 + \ldots + u_N\boldsymbol{a}_N, \quad u_i \in \mathbb{Z}, \quad i = 1, \ldots, N\}. \tag{2.7}$$

The set of vectors $\{\boldsymbol{a}_i\}_{i=1}^N$ are the generator (basis) vectors of the lattice and the matrix $\boldsymbol{M}_\Lambda = (\boldsymbol{a}_1|\boldsymbol{a}_2|\ldots|\boldsymbol{a}_N)^T$ is called the generator matrix of the lattice. Thus, the vertices of the cells C^j form a lattice Λ^j having generator matrix $\boldsymbol{M}_{\Lambda^j} = (\Delta_1^j\boldsymbol{\varphi}_1^j|\Delta_2^j\boldsymbol{\varphi}_2^j)^T$. Because of the orthogonality, the basis vectors of the lattice point in the same directions as the unitary vectors that compose $\boldsymbol{F}^j$, but in general, it is clear that this is not the case when the tight frame is not composed by a set of orthogonal bases, as we will see in section 2.3. There are an infinite number of possible (minimal) bases that can be used for this lattice. We will always use, as a basis for the lattice Λ^j, the j-th orthogonal basis $\{\boldsymbol{\varphi}_1^j, \boldsymbol{\varphi}_2^j\}$. In this way, the outputs of quantizer Q^j can be expressed directly in terms of the generator matrix $\boldsymbol{M}_{\Lambda^j}$. Notice that the cells associated with the quantizer Q^j are convex polytopes whose vertices are all in the lattice Λ^j.

Given a set of quantizers Q^j, $j = 1, \ldots, r$, defined as above, we now introduce the concept of Equivalent Vector Quantizer (EVQ) as follows:

Definition 6 *The EVQ consists of:*

1. *A set of quantization cells formed by the intersection of the rectangular cells $\{C^j\}_{j=1}^r$ of the quantizers $\{Q^j\}_{j=1}^r$.*

2. *A mapping $\mathbb{R}^2 \longrightarrow \mathbb{R}^2$ from the set of cells to a set of reconstructions given by:*

$$\hat{\boldsymbol{x}} = \frac{1}{r} \sum_{j=1}^{r} \hat{\boldsymbol{x}}^j \quad \textit{where} \quad \hat{\boldsymbol{x}}^j = Q^j(\boldsymbol{x}). \tag{2.8}$$

Thus, the linear reconstruction, as represented in Fig. 2.1 and shown in Fig. 2.2(b), consists of taking the geometrical average point among the different reconstructions $\hat{\boldsymbol{x}}^j$, $j = 1, \ldots, r$.

The PSQ in $\mathbb{R}^M$ leads to an EVQ in $\mathbb{R}^2$ and the output of the EVQ can be written as a linear combination of the outputs from each $2D$ quantizer Q^j where it can be seen that the set of outputs (reconstructions) of quantizer Q^j forms a coset of the lattice Λ^j. Fig. 2.2(a) illustrates the partition generated by the EVQ for an example where $r = 2$, and the tight frame and associated stepsizes are:

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \cos(\frac{\pi}{6}) & \sin(\frac{\pi}{6}) \\ -\sin(\frac{\pi}{6}) & \cos(\frac{\pi}{6}) \end{pmatrix} \qquad \Delta_2^1 = \frac{6}{5}\Delta_1^1 \quad \Delta_1^2 = \frac{13}{10}\Delta_1^1 \quad \Delta_2^2 = \frac{9}{8}\Delta_1^1 \tag{2.9}$$

Fig. 2.2(b) illustrates how the final reconstruction vector $\hat{\boldsymbol{x}}$ is obtained. Notice that since the cells C^j generated by the quantizer Q^j are convex polytopes, the cells $C^{EVQ} = C^{\Lambda^1} \cap \ldots \cap C^{\Lambda^r}$ corresponding to the EVQ are intersections of convex polytopes, and therefore are also convex polytopes in $\mathbb{R}^2$. It is important to notice that in general the EVQ is not necessarily a *Voronoi or nearest neighbour* vector quantizer, and although its cells are convex polytopes, they are not in general (minimum distance) Voronoi cells. For a cell to be a Voronoi cell, it would be required that any point contained in that cell be closer to the centroid of that cell than to the centroid of any other cell. This is not satisfied in general because these cells are obtained as the intersection of cells of the (nearest neighbour) quantizers $\{Q^j\}_{j=1}^r$ used in each of the basis, rather than as the nearest neighbor regions for each reconstruction vector. In other words, the intersection of nearest neighbour quantizers does not result in general in a nearest neighbour quantizer. Therefore, we will refer to EVQ cells instead of Voronoi cells. In general, for a given redundancy r, $\hat{\boldsymbol{x}}$ is obtained by averaging over the r linear reconstructions $\hat{\boldsymbol{x}}^j$ $j = 1, \ldots, r$ given by the corresponding quantizers Q^j $j = 1, \ldots, r$.

Remark: The concept of EVQ can be actually defined for any reconstruction algorithm, not necessarily only for the linear reconstruction algorithm using the minimal dual frame as described above. However, for clarity, we have restricted in this section the definition and concepts to this particular case. For any other reconstruction algorithm, the definitions 4 and 6 should be modified so that the set of reconstruction vectors are the ones given by the particular reconstruction algorithm that is used.

Another concept that will be used in some of the next sections is that of fundamental polytope. The fundamental polytope C_o^j associated with the lattice Λ^j is defined by:

$$C_o^j = \{\boldsymbol{x} : \boldsymbol{x} = \alpha_1 \Delta_1^j \boldsymbol{\varphi}_1^j + \alpha_2 \Delta_2^j \boldsymbol{\varphi}_2^j, \qquad 0 \leq \alpha_i < 1, \qquad i = 1, 2\} \tag{2.10}$$

which is the parallelopiped formed by the basis vectors of the lattice Λ^j. The area of this fundamental polytope is equal to $|det(\boldsymbol{M}_{\Lambda^j})|$.

2.2.3 Property of Consistency for a generic reconstruction algorithm

Although the concept of consistency was introduced in [84, 111], for the sake of clarity and because it is a central concept for this work, we review it here. Given a tight frame $\boldsymbol{\Phi}$, constructed by using $r > 1$ orthogonal bases, it is desirable to design an EVQ, such that if $\boldsymbol{x}$ is the original vector and $\hat{\boldsymbol{x}}$ is the reconstructed vector, both $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ fall in the same EVQ cell. The reconstruction vectors $\hat{\boldsymbol{x}}$ satisfying this property are called consistent reconstructions of $\boldsymbol{x}$.

Given a frame operator F and a generic PSQ, the concepts of consistency (for a *generic* reconstruction algorithm) and linear consistency (for the case of *linear* reconstruction) for an EVQ cell C_i^{EVQ}, are defined as follows:

Definition 7 *Consistent cell: Let C_i^{EVQ} be a cell in an EVQ, and $\hat{\boldsymbol{x}}$ its reproduction vector. C_i^{EVQ} is said to be consistent if $\hat{\boldsymbol{x}} \in C_i^{EVQ}$.*

For the particular case of using a linear reconstruction, the definition of linearly consistent cell is:

Definition 8 *Linearly Consistent cell : Let C_i^{EVQ} be a cell of an EVQ. C_i^{EVQ} is said to be linearly consistent if it is consistent under linear reconstruction, where the linear reproduction vector is given by $\hat{\boldsymbol{x}} = \frac{1}{r}\sum_{j=1}^{r} \hat{\boldsymbol{x}}^j$.*

Remark: As before, the definition of linear consistency can be extended to any general linear reconstruction algorithm, not just the linear reconstruction given by the minimal dual frame.

An EVQ is said to be consistent if and only if all its cells C^{EVQ} are consistent. Similarly, a general reconstruction algorithm that gives rise to a consistent quantizer is called a consistent reconstruction algorithm. In particular, a quantizer which satisfies consistency under linear reconstruction is said to be linearly consistent.

Given an EVQ, the optimal reconstruction for any cell in terms of average MSE is obviously inside that cell, that is, the optimal reconstruction is always a consistent reconstruction[2]. Since an inconsistent reconstruction $\hat{\boldsymbol{x}}$ is outside the cell corresponding to the original signal $\boldsymbol{x}$, as opposed to a consistent reconstruction, consistent reconstructions will yield smaller squared distortion (MSE) than inconsistent reconstructions on average for a given EVQ. In our work, the goal is to find a set of EVQ's for which it is possible to have consistent reconstructions with simple reconstruction algorithms.

Fig. 2.2(b) shows examples of both consistent and inconsistent cells assuming linear reconstruction. One of our goals is to design quantization techniques such that all EVQ cells are linearly consistent. Fig. 2.3(a) and Fig. 2.3(b) provide a simple and intuitive example that illustrates how linearly consistent EVQ cells

[2] This statement holds because the EVQ cells are convex.

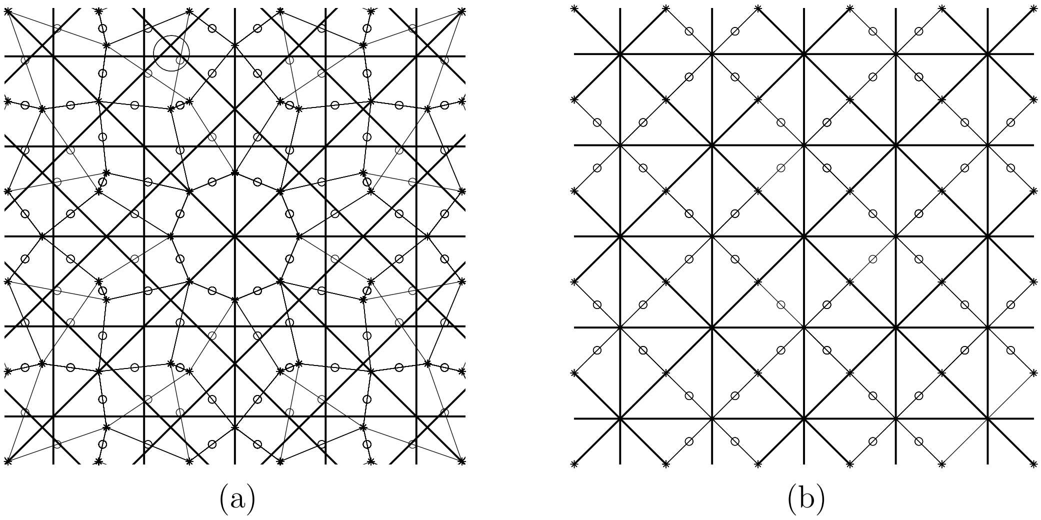
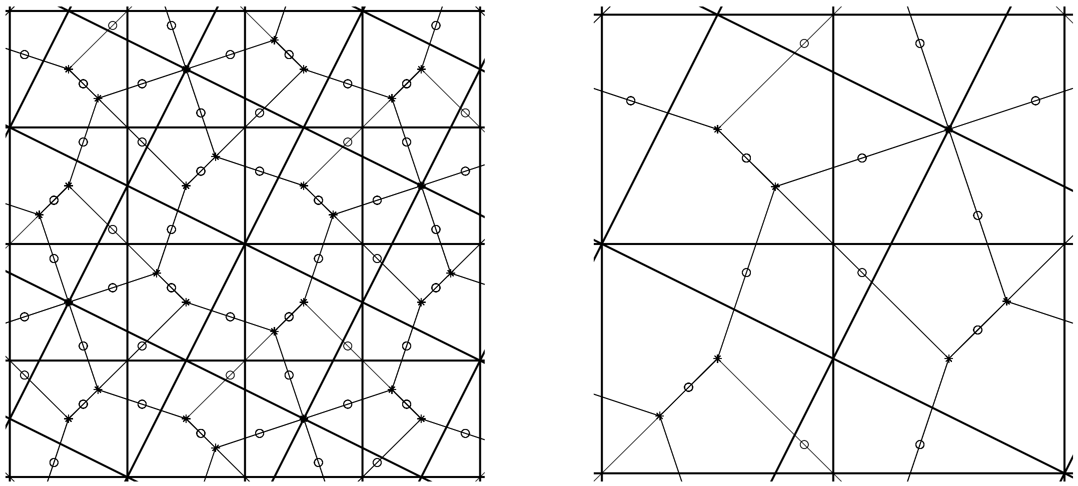


Figure 2.3: Example for $r = 2$ showing how the consistency problem can be solved by choosing carefully a certain frame and a set of different stepsizes: a) using the same stepsizes gives rise to inconsistent cells, one of them is indicated with a circle; b) choosing different stepsizes in each basis yields a consistent EVQ.

can be achieved *by choosing scalar quantizers with different stepsizes for each of the* $r = 2$ *bases.* It can be seen in Fig. 2.3(b) how the intersection between cells of Q^1 and cells of Q^2 is the same across all the partition of the EVQ. As will be explained later, the crucial idea we use in order to achieve consistency with low complexity reconstruction algorithms is to enforce a *periodic* structure on the partition defined by the EVQ, as in the example of Fig. 2.3. Intuitively, the stepsizes selected will depend on the angle between each of the bases. Fig. 2.4 shows a second example where consistency is achieved by creating a periodic structure. Therefore, since it seems intuitive that periodicity in the structure may be useful to remove inconsistency, we analyze next how to construct periodic EVQ's.



(a) (b)

Figure 2.4: a)Example of a linearly consistent quantizer EVQ, b)(Zoom) 4 cells of Q^1. The reconstructions $\hat{\boldsymbol{x}}^j$, $j = 1, 2$ are represented by $'*'$ and the final reconstruction $\hat{\boldsymbol{x}}$ is represented by $'\circ'$.

2.3 Construction and Design of Quantizers with Periodic Structure

We call the type of quantizers shown in Fig. 2.3 “periodic quantizers” because the partition they generate has a periodic structure. We derive in detail how to design such quantizers in this section. The construction that we give in order to achieve periodicity is completely general. However, we provide designs only for redundant families (frames) of vectors with a certain constrained structure. More specifically, we give designs mostly for the case of having r orthogonal bases in $\mathbb{R}^2$. Some designs extensions for $\mathbb{R}^2$ are given in section 2.3.5 where several examples are given, and extensions to higher dimensions can be found in section 2.3.4 and Chapter 3.

2.3.1 Definition and Construction of periodic EVQ's for $\mathbb{R}^2$

In order to facilitate the understanding, we first provide a detailed derivation of how to impose a periodic structure in EVQ's in $\mathbb{R}^2$ for the case of redundancy $r = 2$. Then, we extend the idea to higher redundancies also in $\mathbb{R}^2$, and, finally, we explain how to obtain periodic structures in higher dimensions, showing that the construction is completely general.

In designing an EVQ with a periodic structure, we will use the concept of sublattice.

Definition 9 *([34]) A sublattice $\Lambda_s \in \Lambda$ of a given lattice Λ is a subset of the elements of Λ that is itself a lattice. A sublattice Λ_s is completely specified by an invertible integer matrix $\boldsymbol{B}_{\Lambda_s}$ that maps a basis of Λ into a basis of Λ_s, that is, $\boldsymbol{M}_{\Lambda_s} = \boldsymbol{B}_{\Lambda_s}\boldsymbol{M}_{\Lambda}$, where $\boldsymbol{M}_{\Lambda_s}$ and $\boldsymbol{M}_{\Lambda}$ are the generator matrices of Λ_s and Λ respectively.*

Given a real full rank lattice[3] Λ with generator matrix $\boldsymbol{M}_{\Lambda}$, we consider only full rank sublattices Λ_s, that is, $rank(\boldsymbol{M}_{\Lambda}) = rank(\boldsymbol{M}_{\Lambda_s})$. Another important

[3] Λ is said to be a full rank lattice if its generator matrix $\boldsymbol{M}_{\Lambda}$ is full rank.

concept is that of the index of a sublattice Λ_s contained in a lattice Λ which is given by:

$$|\Lambda/\Lambda_s| = \frac{det(\boldsymbol{M}_{\Lambda_s})}{det(\boldsymbol{M}_{\Lambda})} = \frac{Vol(C_o^{\Lambda_s})}{Vol(C_o^{\Lambda})} = |det(\boldsymbol{B}_{\Lambda_s})| \tag{2.11}$$

The index of a sublattice is the ratio of the volumes of the fundamental polytope associated with the sublattice Λ_s and the one associated with Λ. This is also equal to the number of lattice points of Λ contained in each cell defined by Λ_s. Notice that in the particular case of having an integer matrix $\boldsymbol{B}_{\Lambda_s}$ such that $|det(\boldsymbol{B}_{\Lambda_s})| = 1$, Λ and Λ_s are the same lattice. This particular type of integer matrices satisfying this property are called *unimodular* matrices and by taking different unimodular matrices one can obtain different generator matrices for the same lattice.

We introduce the concept of geometrically scaled-similar sublattices from which we build periodic tesselations.

Definition 10 *Given a real lattice Λ in $\mathbb{R}^2$ with generator matrix $\boldsymbol{M}_\Lambda$, a lattice Λ' is geometrically scaled-similar to Λ iff:*

$$\boldsymbol{M}_{\Lambda'} = \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} \boldsymbol{U}\boldsymbol{M}_\Lambda \boldsymbol{R}, \tag{2.12}$$

where $\boldsymbol{R}$ is a 2×2 orthogonal matrix, that is, a rotation and/or a reflection in $\mathbb{R}^2$, $\boldsymbol{U}$ is a 2×2 unimodular integer matrix, and $c_1, c_2 \in \mathbb{R}_+$.

If Λ' is geometrically scaled-similar to Λ and is also a sublattice of Λ, then we denote it by $S\Lambda$. Note that this can only be true for specific values of c_1, c_2 and $\boldsymbol{R}$. Thus, a geometrically scaled-similar sublattice $S\Lambda$ of a lattice Λ is obtained by simply rotating and/or reflecting the lattice Λ and then scaling each of the new axes. The matrix $\boldsymbol{U}$ allows us to choose different basis vectors for the sublattice $S\Lambda$. If $det(\boldsymbol{R}) = +1$, then $\boldsymbol{R}$ is a pure rotation, and the scaling parameters c_1 and c_2 allow to control the magnitudes in each of the 2 vectors that define its basis. If $det(\boldsymbol{R}) = -1$, then $\boldsymbol{R}$ contains or is a reflection. The possible orientations and values for c_1 and c_2 that determine a geometrically scaled-similar sublattice will be given in section 2.3.2. Notice that in the particular case of having $c_1 = c_2$, $S\Lambda$ would be a geometrically similar (or equivalent) sublattice of Λ, as defined by Conway et al. [30, 34]. We restrict $\boldsymbol{R}$ to be a pure rotation

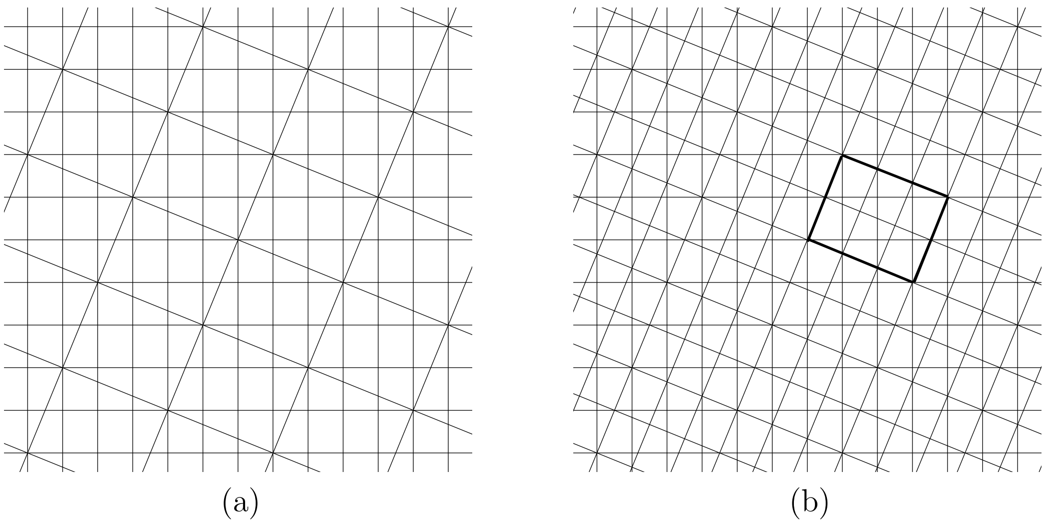


Figure 2.5: Example 1: a) Sublattice structure b) EVQ cells C^{EVQ}.

so that we can associate each rotation with a basis of a frame, as we explain next. Fig. 2.5(a) shows an example for a redundancy $r = 2$ of a geometrically scaled-similar sublattice of a rectangular lattice.

Without loss of generality, in the following we will construct geometrically scaled-similar sublattices of a canonical lattice Λ^1, where Λ^1 has generator matrix:

$$\boldsymbol{M}_{\Lambda^1} = \begin{pmatrix} \Delta_1^1 & 0 \\ 0 & \Delta_2^1 \end{pmatrix} \tag{2.13}$$

that is, the generator vectors of Λ^1 are scaled versions of the canonical basis vectors $\boldsymbol{\varphi}_1^1 = [1, 0]^T$, $\boldsymbol{\varphi}_2^1 = [0, 1]^T$ ($\boldsymbol{F}^1 = \boldsymbol{I}_{2\text{x}2}$). We define the quantizer Q^1 as the quantizer with rectangular cells C^{Λ^1} whose vertices are given by the lattice Λ^1.

Notational Remark: In order to distinguish between the cells associated with a lattice Λ^j or a quantizer Q^j and the cells associated with a geometrically scaled-similar sublattice $S\Lambda^j \subset \Lambda^j$, we will use the following notation: a) C^{Λ^j} will denote the set of cells associated with Λ^j and Q^j, where we use now C^{Λ^j} instead of C^j in order to emphasize that these cells are associated with the lattice Λ^j; b) $C^{S\Lambda^j}$ will denote the set of cells associated with $S\Lambda^j$. The subscript will indicate in both cases a particular cell.

Definition 11 *Periodicity Property: An EVQ is said to be periodic if the partition of the space given by its quantizing cells satisfies the following two properties:*

1. *There exists a minimal periodic unit C_o^{EVQ} which is the union of a finite set of cells $\{\mathcal{P}_1, \ldots, \mathcal{P}_m\}$.*

2. *There exists a lattice Λ which determines this periodicity such that all the cells of the EVQ are given by $\{\mathcal{P}_1, \ldots, \mathcal{P}_m\} + \Lambda$, that is, copies of the minimal unit C_o^{EVQ} translated by the points of Λ.*

Fig. 2.5(b) shows the unit cell C_o^{EVQ} with bold lines for a particular EVQ with redundancy $r = 2$.

The periodicity structure is achieved by finding lattices whose intersection is not empty, which involves the concept of sublattice.

Fact 1 *If Λ_s is a sublattice of Λ^1, the partition defined by the intersection of the cells C^{Λ^1} with the cells determined by Λ_s has a periodic structure (tesselation) with the minimal periodic unit given by $C^{\Lambda^1} \cap C_o^{\Lambda_s}$, where $C_o^{\Lambda_s}$ is the fundamental polytope associated with the sublattice Λ_s and the whole tesselation is obtained by translating the cells $C^{\Lambda^1} \cap C_o^{\Lambda_s}$ with the points of Λ_s.*

Proof: See Appendix A.1.

This fact can be observed in Fig. 2.5(a) where in this case, the sublattice is a geometrically scaled-similar sublattice. In this work, we use Fact 1 for the particular case where the sublattices are geometrically-scaled similar.

Definition 12 *Given a set of lattices Λ^j, $j = 1, \ldots, r$, the coincidence site lattice (CSL) Λ^{CSL} is the intersection lattice:*

$$\Lambda^{CSL} = \Lambda^1 \cap \Lambda^2 \cap \ldots \cap \Lambda^r, \tag{2.14}$$

which is the finest common sublattice of all the lattices Λ^j, $j = 1, \ldots, r$.

In order to achieve periodicity, our goal is to construct a set of lattices Λ^1, Λ^2,..., Λ^r whose intersection is not empty. For this, it is sufficient to find a set of geometrically scaled-similar sublattices $S\Lambda^1$, $S\Lambda^2$,..., $S\Lambda^r$ of the first lattice Λ^1. For notational convenience, we take $S\Lambda^1 = \Lambda^1$ and we will always take $\boldsymbol{U} = \boldsymbol{I}$ in (2.12) so that the basis vectors of the j-th geometrically scaled-similar sublattice are orthogonal (because of the rotation matrix) and can be associated with the j-th orthogonal basis of a tight frame. Each rectangular cell $C_i^{S\Lambda^j}$ defined by each sublattice $S\Lambda^j$ has sides with lengths $c_1^j\Delta_1^1$ and $c_2^j\Delta_2^1$. Since we have that $c_1^j, c_2^j \geq 1 \; \forall j$, $Vol(S\Lambda^j) = c_1^j c_2^j Vol(\Lambda^1) \geq Vol(\Lambda^1)$. Moreover, since the index of a sublattice is always an integer, we have that $c_1^j \times c_2^j \in \mathbb{Z} \; \forall j$.

Suppose we design jointly a lattice $\Lambda^1 = S\Lambda^1$ with generator matrix $\boldsymbol{M}_{\Lambda^1} = diag[\Delta_1^1, \Delta_2^1]$ (choosing certain values for Δ_1^1, Δ_2^1), and $r-1$ different geometrically scaled-similar sublattices of Λ^1 denoted by $S\Lambda^2$, $S\Lambda^3$,..., $S\Lambda^r$. Given a sublattice $S\Lambda^j$, we define a finer lattice $\Lambda^j \supset S\Lambda^j$ with generator matrix given by:

$$\boldsymbol{M}_{\Lambda^j} = \begin{pmatrix} \frac{1}{d_1^j} & 0 \\ 0 & \frac{1}{d_2^j} \end{pmatrix} \boldsymbol{M}_{S\Lambda^j},\ \boldsymbol{M}_{S\Lambda^j} = \boldsymbol{B}_{S\Lambda^j} \begin{pmatrix} \Delta_1^1 & 0 \\ 0 & \Delta_2^1 \end{pmatrix},\ \boldsymbol{B}_{S\Lambda^j} = \begin{pmatrix} k_{11}^j & k_{12}^j \\ -k_{21}^j & k_{22}^j \end{pmatrix} \tag{2.15}$$

where d_1^j, d_2^j, k_{11}^j, k_{12}^j, k_{21}^j, $k_{22}^j \in \mathbb{Z}_+$, that is, are any positive integers.

As we show in Lemma 1 below, if we associate r quantizers $\{Q^j\}_{j=1}^r$ respectively with the lattices $\{\Lambda^j\}_{j=1}^r$, this construction given above is sufficient in order to ensure that the intersection of all the lattices Λ^j, $j = 1, \ldots, r$ is not empty, and therefore, by group theory, the intersection is a lattice. Notice that if we consider only one lattice Λ^j together with the canonical lattice Λ^1, both constructed as described in (2.15), and we define corresponding quantizers Q^1 and Q^j, respectively associated with them, it follows from Fact 1 and because $\{d_1^j, d_2^j\}$ are positive integers, that the cells given by $C^{\Lambda^1} \cap C^{\Lambda^j}$ have a periodic structure, which is still determined by $C_o^{S\Lambda^j}$ (see Fig. 2.5(b)). Therefore, for $r = 2$, it is clear that periodicity holds.

Next, we show that the construction of $\Lambda^1, \ldots, \Lambda^r$ given above ensures that these lattices have a non-empty intersection, which actually implies a periodic structure[4] in the resulting EVQ.

[4] Notice that a periodic tesselation may be obtained also using other methods which are not based on intersecting lattices, that is, forcing the intersection of the lattices $\Lambda^1, \ldots, \Lambda^r$ is just one (purely geometrical) way to obtain a periodic tesselation, but one could also build a periodic tesselation in other ways.

Lemma 1 *Given a set of lattices* $\{\Lambda^j\}_{j=2}^{j=r}$*, such that* $\Lambda^j \supset S\Lambda^j$ *and* $S\Lambda^j$ *is a sublattice of* Λ^1 $j = 2, \ldots, r$*, then the coincidence site lattice contains as a sublattice, a lattice* Λ^o *that is an integer scaling of* Λ^1*, that is,* $\boldsymbol{M}_{\Lambda^o} = D\boldsymbol{M}_{\Lambda^1}$*, where* $D \in \mathbb{Z}$*.*

Proof: See Appendix A.2. The importance of calculating the coincidence site lattice Λ^{CSL} comes from the fact that its fundamental cell C_o^{CSL} is the unit cell that is repeated in the periodic structure of the resulting EVQ, as shown in the following Lemma.

Lemma 2 *Given* r *quantizers* Q^j*,* $j = 1, \ldots, r$*, associated with the lattices* Λ^j*,* $j = 1, \ldots, r$*, the partition of* EVQ *cells has a periodic structure, with the unit cell that is repeated periodically being* C_o^{CSL}*, the fundamental polytope of the coincidence site lattice* Λ^{CSL}*.*

Proof: See Appendix A.3.

Notice that any other lattice that is also a sublattice (although coarser than the CSL) of all the lattices Λ^j, $j = 1, \ldots, r$ determines also a unit cell that is repeated periodically. This unit cell, however, will be larger than C_o^{CSL}. For instance, the fundamental polytope of the rectangular lattice Λ^o described in Lemma 1, will be also repeated periodically but $Vol(\Lambda^o) \geq Vol(\Lambda^{CSL})$.

Next, we show how to calculate in a simple way the generator matrix of the coincidence site lattice Λ^{CSL} for any dimension N. For this, it is necessary to first review the following concept for N-dimensional lattices.

Definition 13 *Given r N-dimensional lattices Λ^j, $j = 1, \ldots, r$ in $\mathbb{R}^N$ satisfying the property that $\exists$ an N-dimensional lattice Λ^F for which $\Lambda^j \subset \Lambda^F$, $j = 1, \ldots, r$, we define the (N-dimensional) sum lattice $\Lambda^\Sigma = \Lambda^1 + \Lambda^2 + \ldots \Lambda^r$ as follows [83]:*

$$\Lambda^\Sigma = \{\boldsymbol{y} \in \mathbb{R}^N : \boldsymbol{y} = \boldsymbol{x}\boldsymbol{A}, \boldsymbol{x} \in \mathbb{Z}^{rN}\} \quad \textit{where} \qquad \boldsymbol{A} = \begin{pmatrix} \boldsymbol{M}_{\Lambda^1} \\ \boldsymbol{M}_{\Lambda^2} \\ \vdots \\ \boldsymbol{M}_{\Lambda^r} \end{pmatrix}. \tag{2.16}$$

Remark: The lattice Λ^Σ is the lattice generated by all the basis vectors of all the lattices Λ^j, $j = 1, \ldots, r$ in $\mathbb{R}^N$ (not simply the union of the lattice points). The matrix $\boldsymbol{A}$ defined above can be reduced to obtain the actual ($N \times N$) generator matrix $\boldsymbol{M}_{\Lambda^\Sigma}$ using the so-called Hermite normal form (HNF) reduction algorithm [83].

Definition 14 *The dual lattice Λ^* of a lattice Λ in $\mathbb{R}^N$ is defined as follows [34]:*

$$\Lambda^* = \{\boldsymbol{v} \in \mathbb{R}^N : \langle \boldsymbol{v}, \boldsymbol{w} \rangle \in \mathbb{Z} \quad \forall \boldsymbol{w} \in \Lambda\}. \tag{2.17}$$

The generator matrix of Λ^* is given by $\boldsymbol{M}_{\Lambda^*} = ((\boldsymbol{M}_\Lambda)^{-1})^T$, and we have also that $(\Lambda^*)^* = \Lambda$ [34].

It is important to note that the sum of 2 lattices Λ^1 and Λ^2 is not necessarily a lattice; for instance taking $\Lambda^1 = \mathbb{Z}$ and $\Lambda^2 = \sqrt{2}\mathbb{Z}$, then their sum is not a lattice because the sum is not a discrete subgroup of $\mathbb{R}$. It can be shown [47, 78] that if Λ^1 and Λ^2 are contained in a certain full rank lattice Λ^F, then $\Lambda^1 + \Lambda^2$ is a full rank lattice. There exists a fast algorithm to find the basis of the sum of two lattices which makes use of the concept of greatest common left divisor (gcld) of two matrices (this can be found in Appendix A.8).

Based on the previous definitions, the following important theorem from lattice theory allows us to calculate the intersection lattice Λ^{CSL} of a set of lattices $\Lambda^1, \ldots, \Lambda^r$ [74, 87]:

Theorem 1 *Given r lattices Λ^j, $j = 1, \ldots, r$, the following holds:*

$$(\Lambda^1)^* + \ldots + (\Lambda^r)^* = (\Lambda^1 \cap \ldots \cap \Lambda^r)^* \Longleftrightarrow \left((\Lambda^1)^* + \ldots + (\Lambda^r)^*\right)^* = \Lambda^1 \cap \ldots \cap \Lambda^r \ . \tag{2.18}$$

Notice that using Lemma 1 the construction of the lattices $\Lambda^1, \ldots, \Lambda^r$ we have presented here ensures that Λ^{CSL} always exist and is a full rank lattice, implying a periodic structure in the EVQ. The necessary and sufficient condition for $\Lambda^1 \cap \Lambda^2$ to exist and be a full rank lattice is that the matrix $(M_{\Lambda^1})^{-1} M_{\Lambda^2}$ be a matrix of rational numbers. This condition is implicitly used in order to prove Lemma 1. In the same way, our construction also ensures that $(\Lambda^1)^* + (\Lambda^2)^*$ always exists and is a full rank lattice. The lattice $\Lambda^1 \cap \Lambda^2$ is the finest lattice which is a sublattice of Λ^1 and Λ^2, while the sum $\Lambda^1 + \Lambda^2$ is the coarsest lattice which contains both Λ^1 and Λ^2 as sublattices.

2.3.2 Design and Parameterization for $\mathbb{R}^2$

Let Λ^1 be a rectangular lattice in $\mathbb{R}^2$ with generator matrix $\boldsymbol{M}_{\Lambda^1} = diag[\Delta_1^1, \Delta_2^1]$, which defines a quantizer Q^1. In $\mathbb{R}^2$, it is easy to parameterize all the geometrically scaled-similar sublattices of Λ^1 in terms of the possible *scaling factors* and *rotation* matrices as in (2.12). This parameterization can be used in order to build a periodic EVQ in $\mathbb{R}^2$ for any redundancy r.

Fact 2 *All geometrically scaled-similar sublattices $S\Lambda$ of Λ^1 with $\boldsymbol{M}_{\Lambda^1} = diag[\Delta_1^1,$ $\Delta_2^1]$, have generator matrices that can be characterized geometrically in the following way:*

$$\boldsymbol{M}_{S\Lambda} = \begin{pmatrix} c_1\Delta_1^1 & 0 \\ 0 & c_2\beta\Delta_1^1 \end{pmatrix} \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix}$$

$$where \quad \beta = \frac{\Delta_2^1}{\Delta_1^1} = \sqrt{\frac{k_{11}k_{21}}{k_{12}k_{22}}}, \quad \tan(\theta) = \sqrt{\frac{k_{12}k_{21}}{k_{11}k_{22}}} = \frac{k_{12}}{k_{11}}\beta, \quad c_1 = \frac{k_{11}}{cos(\theta)}, \quad c_2 = \frac{k_{22}}{cos(\theta)} \tag{2.19}$$

and k_{11}*,* k_{12}*,* k_{21}*,* k_{22} *are any possitive integers and* $0 < \theta < \frac{\pi}{2}$.

Proof: See Appendix A.4.

The angle θ is restricted to the interval $]0, \frac{\pi}{2}[$ to avoid duplicity. That is, given a valid angle $\theta \in]0, \frac{\pi}{2}[$, the angles $\theta + i\frac{\pi}{2}$, $i = 1, 2, 3$ generate the same sublattice $S\Lambda$ because the basis vectors will be inverted versions of the ones corresponding to $\theta \in]0, \frac{\pi}{2}[$.

The generator matrix of lattice Λ^j, as given in (2.15), and stepsizes $\{\Delta_1^j, \Delta_2^j\}$ associated with the scalar quantizers $\{SQ_1^j, SQ_2^j\}$ can be parameterized by:

$$M_{\Lambda^j} = \begin{pmatrix} \frac{k_{11}^j}{d_1^j} & \frac{k_{12}^j}{d_1^j}\beta \\ \frac{-k_{21}^2}{d_2^j} & \frac{k_{22}^j}{d_2^j}\beta \end{pmatrix} \Delta_1^1, \qquad \begin{array}{l} \Delta_1^j = \frac{\Delta_1^1}{d_1^j}\sqrt{\frac{k_{11}^j}{k_{22}^j}(k_{11}^j k_{22}^j + k_{12}^j k_{21}^j)} \\ \Delta_2^j = \frac{\Delta_1^1}{d_2^j}\sqrt{\frac{k_{21}^j}{k_{12}^j}(k_{11}^j k_{22}^j + k_{12}^j k_{21}^j)} \end{array} \tag{2.20}$$

A few comments are in order:

1. Only those angles θ such that $\tan(\theta) = \sqrt{m_1/m_2}$, $m_1, m_2 \in \mathbb{Z}_+$, lead to geometrically scaled-similar sublattices.

2. For a given fixed angle θ there is more than one solution for β, c_1 and c_2.

3. The product $c_1 c_2 = |\Lambda/S\Lambda| \in \mathbb{Z}_+$, as it should be, because:

$$c_1 c_2 = k_{11}k_{22}\left(\frac{1}{\cos(\theta)}\right) = k_{11}k_{22}(1 + (\tan(\theta))^2) = k_{11}k_{22} + k_{12}k_{21} \in \mathbb{Z}_+ \tag{2.21}$$

4. If we consider the particular case of having $c_1 = c_2 = c$ and $\beta = 1$, that is, geometrically similar sublattices of the cubic real lattice $\mathbb{Z}^2_{\Delta_1^1}$, then, the possible solutions are[5]:

$$\tan(\theta) = \frac{b}{a}, \quad c = \sqrt{a^2 + b^2}, \quad \cos(\theta) = \frac{a}{\sqrt{a^2 + b^2}}, \quad \sin(\theta) = \frac{b}{\sqrt{a^2 + b^2}} \tag{2.22}$$

   where $a, b \in \mathbb{Z}_+$, which agrees with [30].

Although periodicity in the structure holds for any two positive integers d_1^j and d_2^j, in practice, each pair (d_1^j, d_2^j) is constrained to some values to provide good quantization performance. Therefore, it is desirable not to have a cell of a quantizer Q^{j_1} completely contained within a cell of another quantizer Q^{j_2}. Ideally, adding succesive quantizers Q^j will lead to reductions in the size of the EVQ cells (and therefore in distortion). Appendix A.5 describes in detail a simple geometric criterion that can be used to address this issue. There is not a unique way in the order in which one can choose the different parameters. One possible way is by fixing the angle θ first, that is, choosing a value for $\sqrt{(k_{12}k_{21})/(k_{11}k_{22})}$, then searching within all the 4-tuples of integers resulting in that value, and for each of these 4-tuples, we obtain certain values for the stepsizes using (2.20).

[5] Notice that we are restricting the angle θ to be $0 < \theta < \frac{\pi}{2}$.

2.3.3 Examples of Periodic EVQ's in $\mathbb{R}^2$

We present in this section several design examples for the two-dimensional case.

Example 1 *Let us choose an angle θ such that $\tan(\theta) = \sqrt{2 \times 3}$. A possible choice for the constant integers is $k^2_{11} = k^2_{22} = 1$, $k^2_{12} = 2$ and $k^2_{21} = 3$. If we choose $d^2_1 = 2$ and $d^2_2 = 3$, the resulting quantizer Q^2 is given by:*

$$\beta = \sqrt{\frac{3}{2}}, \quad \Delta^2_1 = \sqrt{\frac{3}{2}}\Delta^1_1, \quad \Delta^2_1 = \frac{1}{\cos(\theta)}\Delta^1_1, \quad \Delta^2_2 = \frac{1}{\cos(\theta)}\sqrt{\frac{3}{2}}\Delta^1_1 \tag{2.23}$$

The corresponding EVQ cells are shown in Fig. 2.5(b).

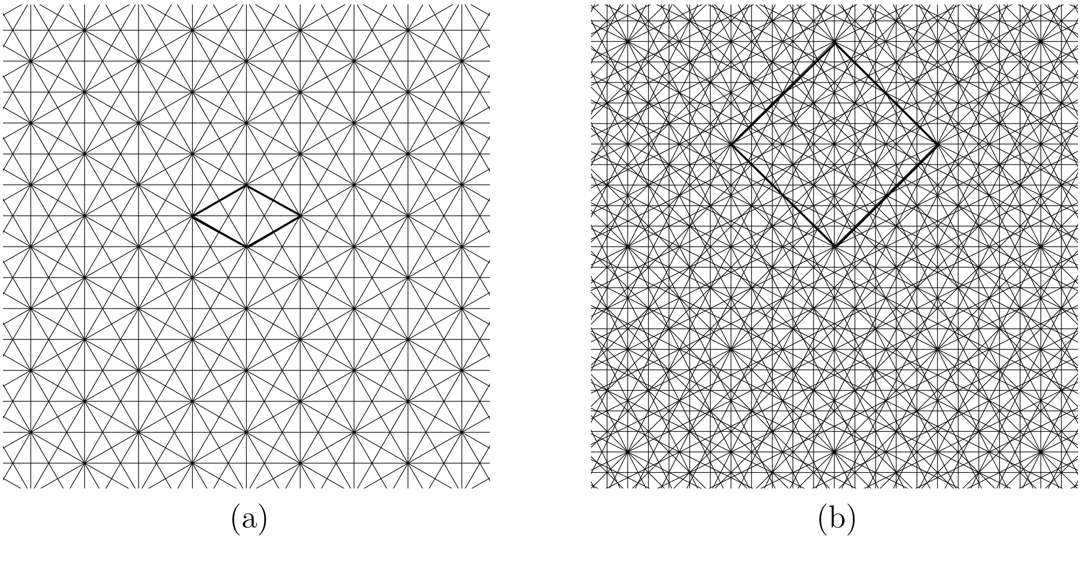


Figure 2.6: a) Example for $r = 3$: Structure of the EVQ and unit cell of the structure; b) Example for $r = 4$: Structure of the EVQ and unit cell. Notice that in b) due to the symmetry that exists within C_o^{CSL}, the effective number of different EVQ cells is basically 1/8 of the total number of cells within this unit cell.

Example 2 *A good example for $r = 3$ is obtained by using the following tight frame and stepsizes:*

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \cos(\frac{\pi}{6}) & \sin(\frac{\pi}{6}) \\ -\sin(\frac{\pi}{6}) & \cos(\frac{\pi}{6}) \\ \cos(\frac{\pi}{3}) & \sin(\frac{\pi}{3}) \\ -\sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) \end{pmatrix} \qquad \begin{array}{ll} \beta = \frac{1}{\sqrt{3}}, & \Delta_2^1 = \beta \Delta_1^1 = \frac{1}{\sqrt{3}} \Delta_1^1 \\ \Delta_1^2 = \frac{1}{2}\left(\frac{1}{\cos(\frac{\pi}{6}}\right)\Delta_1^1, & \Delta_2^2 = \frac{1}{2}\left(\frac{3}{\cos(\frac{\pi}{6}}\right)\left(\frac{1}{\sqrt{3}}\right)\Delta_1^1 \\ \Delta_1^3 = \frac{1}{2}\left(\frac{1}{\cos(\frac{\pi}{3}}\right)\Delta_1^1, & \Delta_2^3 = \frac{1}{2}\left(\frac{1}{\cos(\frac{\pi}{3}}\right)\left(\frac{1}{\sqrt{3}}\right)\Delta_1^1 \end{array} \tag{2.24}$$

Notice that in this example, $d_1^2 = d_2^2 = d_1^3 = d_2^3 = 2$. Fig. 2.6(a) shows the unit cell that is repeated periodically and the resulting EVQ cells. In this example, we have that:

$$\boldsymbol{M}_{\Lambda^{CSL}} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \boldsymbol{M}_{\Lambda^1} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{3}} \end{pmatrix} \Delta_1^1 = \begin{pmatrix} 1 & \frac{1}{\sqrt{3}} \\ -1 & \frac{1}{\sqrt{3}} \end{pmatrix} \Delta_1^1$$

$$\boldsymbol{M}_{\Lambda^o} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \boldsymbol{M}_{\Lambda^1} = 2 \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{3}} \end{pmatrix} \Delta_1^1 \tag{2.25}$$

Example 3 *An example for $r = 4$ can be obtained by using the following tight frame and stepsizes:*

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{-2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{pmatrix} \qquad \begin{array}{ll} \beta = 1, & \Delta_2^1 = \beta\Delta_1^1 = \Delta_1^1 \\ \Delta_1^2 = \sqrt{2}\Delta_1^1, & \Delta_2^2 = \sqrt{2}\Delta_1^1 \\ \Delta_1^3 = \frac{\sqrt{5}}{2}\Delta_1^1, & \Delta_2^3 = \frac{\sqrt{5}}{2}\Delta_1^1 \\ \Delta_1^4 = \frac{\sqrt{5}}{2}\Delta_1^1, & \Delta_2^4 = \frac{\sqrt{5}}{2}\Delta_1^1 \\ \Delta_1^5 = \frac{\sqrt{5}}{2}\Delta_1^1, & \Delta_2^5 = \frac{\sqrt{5}}{2}\Delta_1^1 \\ \Delta_1^6 = \frac{\sqrt{5}}{2}\Delta_1^1, & \Delta_2^6 = \frac{\sqrt{5}}{2}\Delta_1^1 \end{array} \tag{2.26}$$

Fig. 2.6(b) shows the unit cell that is repeated periodically and the resulting EVQ cells. In this example, we have that:

$$\begin{aligned} \boldsymbol{M}_{\Lambda^{CSL}} &= \begin{pmatrix} -5 & 5 \\ 5 & 5 \end{pmatrix} \boldsymbol{M}_{\Lambda^1} = \begin{pmatrix} -5 & 5 \\ 5 & 5 \end{pmatrix} \Delta_1^1 \\ \boldsymbol{M}_{\Lambda^o} &= \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \boldsymbol{M}_{\Lambda^1} = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \Delta_1^1 \end{aligned} \tag{2.27}$$

Notice in these two examples how we have chosen the stepsizes of the different quantizers $\{Q^j\}_{j=1}^r$ trying to satisfy as much as possible the constraints mentioned in section 2.3.2 (refinement between different quantizers).

2.3.4 Design of Periodic EVQ's in higher dimensions

We now analyze the extension to higher dimensions for the case where $\boldsymbol{M}_{\Lambda^1} = \boldsymbol{I}\Delta_1^1$, that is, if the dimension is N, then $\Delta_1^1 = \Delta_2^1 = \ldots = \Delta_N^1$. Since Λ^1 is a cubic lattice, a geometrically scaled similar sublattice $S\Lambda$ has to be also cubic and thus its generator matrix has to be $\boldsymbol{M}_{S\Lambda} = \boldsymbol{B}_{S\Lambda}\boldsymbol{M}_{\Lambda^1} = \boldsymbol{B}_{S\Lambda}\Delta_1^1$ where the integer matrix $\boldsymbol{B}_{S\Lambda}$ satisfies the orthogonality property:

$$\boldsymbol{B}_{S\Lambda}^T\boldsymbol{B}_{S\Lambda} = \begin{pmatrix} b_1 & 0 & \cdots & 0 \\ 0 & b_2 & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & b_N \end{pmatrix}, \qquad b_1, b_2, \ldots, b_N \in \mathbb{Z}_+ \qquad (2.28)$$

If $S\Lambda^j$ is the j-th sublattice, we construct the j-th lattice Λ^j as we have done before for $N = 2$, that is, dividing by integers $\{d_i^j\}_{i=1}^N$ and the associated orthogonal matrix $\boldsymbol{F}^j$ and stepsizes $\{\Delta_i^j\}_{i=1}^N$ will be given by:

$$\boldsymbol{F}^j = \begin{pmatrix} \frac{1}{\sqrt{b_1^j}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{b_2^j}} & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sqrt{b_N^j}} \end{pmatrix} \boldsymbol{B}_{S\Lambda^j}, \qquad \Delta_i^j = \frac{\sqrt{b_i^j}}{d_i^j}\Delta_1^1, \quad d_i^j \in \mathbb{Z}_+ \tag{2.29}$$

and all the results regarding periodicity in the structure of the final EVQ and the coincidence site lattice Λ^{CSL} apply also here.

Since the matrix $\boldsymbol{M}_{S\Lambda}$ is proportional to $\boldsymbol{B}_{S\Lambda}$ by Δ_1^1, let us focus on the problem of finding integer matrices $\boldsymbol{B}_{S\Lambda}$ satisfying the properties mentioned above, thus, looking at geometrically similar sublattices of $\mathbb{Z}^N$. Clearly, we can construct matrices $\boldsymbol{B}_{S\Lambda}$ in the following way:

$$\boldsymbol{B}_{S\Lambda} = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_N \end{pmatrix} \boldsymbol{H}_{S\Lambda}, \; a_1, \ldots, a_N \in \mathbb{Z}, \; \boldsymbol{H}_{S\Lambda}^T \boldsymbol{H}_{S\Lambda} = m\boldsymbol{I} \tag{2.30}$$

where $m \in \mathbb{Z}_+$. The problem of finding matrices $\boldsymbol{H}_{S\Lambda}$ satisfying the above property has been studied extensively [54, 121], where the algebraic theory of orthogonal designs allows to find general constructions of orthogonal matrices with indeterminate entries.

Notice that the matrices $\boldsymbol{H}_{S\Lambda}$ actually generate geometrically similar or equivalent sublattices with index $K = m^{N/2}$, $m \in \mathbb{Z}_+$. Explicit constructions in higher dimensions have been provided by Sloane and Beferull-Lozano and can be found in [102] and Chapter 3. More specifically, constructions are given for dimensions $N = 3, 6, 12, 24, 2^k,\ k \geq 2$. For illustration purposes, we present here a simple example for $N = 4$. Details about the tesselation of the space that is generated are also given in [102] and Chapter 3.

Example 4

$$\boldsymbol{M}_{\Lambda^1} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \boldsymbol{M}_{\Lambda^2} = \begin{pmatrix} + & + & + & + \\ + & - & + & - \\ + & - & - & + \\ + & + & - & - \end{pmatrix}, \boldsymbol{M}_{\Lambda^3} = \begin{pmatrix} - & + & + & + \\ - & - & + & - \\ - & - & - & + \\ - & + & - & - \end{pmatrix} \tag{2.31}$$

where $+ = +1$ and $- = -1$ and the lattices are given at a fixed scale. These lattices can be scaled as usual multiplying them by Δ_1^1. The intersection of these three lattices, that is, the coincidence site lattice can be easily calculated and is given by:

$$\boldsymbol{M}_{\Lambda^{CSL}} = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}, \tag{2.32}$$

which is a version of the well-known lattice D_4 (best known lattice quantizer in four dimensions) on the scale at which its minimal squared norm is 8.

2.3.5 Design of Periodic EVQ's for other redundant families

It is also possible to construct periodic quantizers using families (frames) of vectors with integer redundancy r but which do not consist of a set of orthogonal bases. In this section, we show examples which are based on hexagonal lattices A_2 in $\mathbb{R}^2$, and sublattices which are geometrically similar ($c_1 = c_2 = c$) to hexagonal lattices.

Conway and Sloane [30] have parameterized all the possible sublattices which are geometrically similar to the hexagonal lattice $\Lambda^1 = A_2$, whose generator matrix is given by:

$$\boldsymbol{M}_{\Lambda^1} = \begin{pmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix} \Delta \tag{2.33}$$

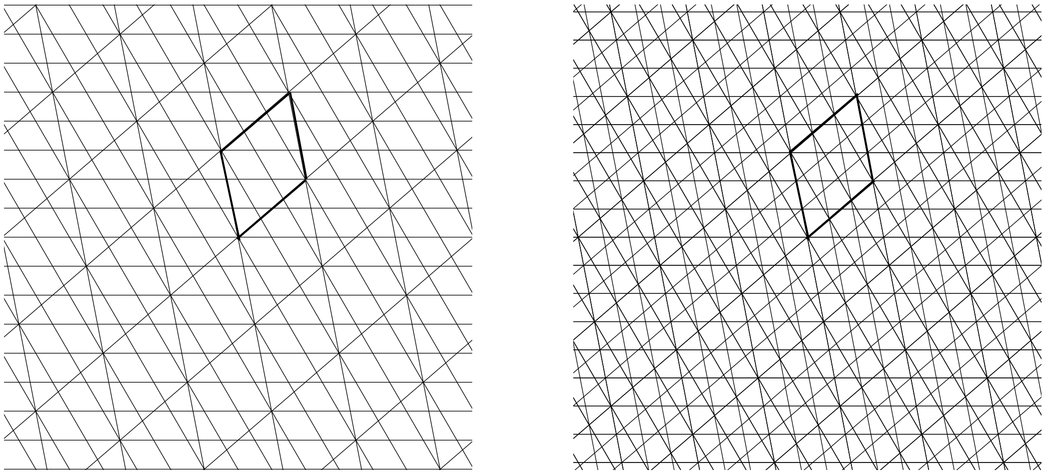
Notice that if we want to associate this lattice with a basis of a frame ($\boldsymbol{F}^1$), the vectors of this basis have to be orthogonal to the basis vectors of the lattice. Moreover, the stepsizes associated with the vectors that compose $\boldsymbol{F}^1$ have to be calculated so that the lines in $\mathbb{R}^2$ intersect exactly to generate M_{Λ^1}. It is trivial to show by simple trigonometry that $\boldsymbol{F}^1$ and the associated stepsizes are:

$$\boldsymbol{F}^1 = \begin{pmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \quad \Delta_1^1 = \Delta_2^1 = \tfrac{\sqrt{3}}{2}\Delta \tag{2.34}$$

It is shown in [30] that a sublattice $S\Lambda$, which is geometrically similar to Λ^1, is generated by (using complex notation) $\boldsymbol{u} = a + b\omega$ and $\boldsymbol{v} = \omega(a + b\omega)$, where $\omega = -1/2 + i\sqrt{3}/2$, $a, b \in \mathbb{Z}$, and the index $|S\Lambda/\Lambda^1|$ of the corresponding sublattice is $|S\Lambda/\Lambda^1| = a^2 - ab + b^2$. Translating this to matrix notation, we have that the possible generator matrices for $S\Lambda$ are given by:

$$\boldsymbol{M}_{S\Lambda} = \begin{pmatrix} \left(a - \frac{b}{2}\right) & \frac{\sqrt{3}}{2}b \\ -\frac{a+b}{2} & \frac{\sqrt{3}}{2}(a-b) \end{pmatrix} \Delta \tag{2.35}$$

Notice also that M_{Λ^1} and $M_{S\Lambda}$ are related as follows:



(a) (b)

Figure 2.7: Example for $r = 2$: a) Structure of the sublattice $S\Lambda$ with $a = 1, b = 3$; b) Structure of the EVQ and unit cell.

$$\boldsymbol{M}_{S\Lambda} = \boldsymbol{M}_{\Lambda^1} \begin{pmatrix} \left(a - \frac{b}{2}\right) & \frac{\sqrt{3}}{2}b \\ -\frac{\sqrt{3}}{2}b & \left(a - \frac{b}{2}\right) \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a-b \end{pmatrix} \boldsymbol{M}_{\Lambda^1} \qquad (2.36)$$

which corresponds to a rotation of an angle θ such that $\tan(\theta) = \frac{\sqrt{3}b}{2a-b}$ and a scaling of $\sqrt{a^2 - ab + b^2}$.

Using this approach, we can design again frames and PSQ's such that a periodic EVQ is generated. Fig. 2.7 and 2.8 show examples of periodic EVQ's for redundancies $r = 2$ and $r = 3$, respectively.

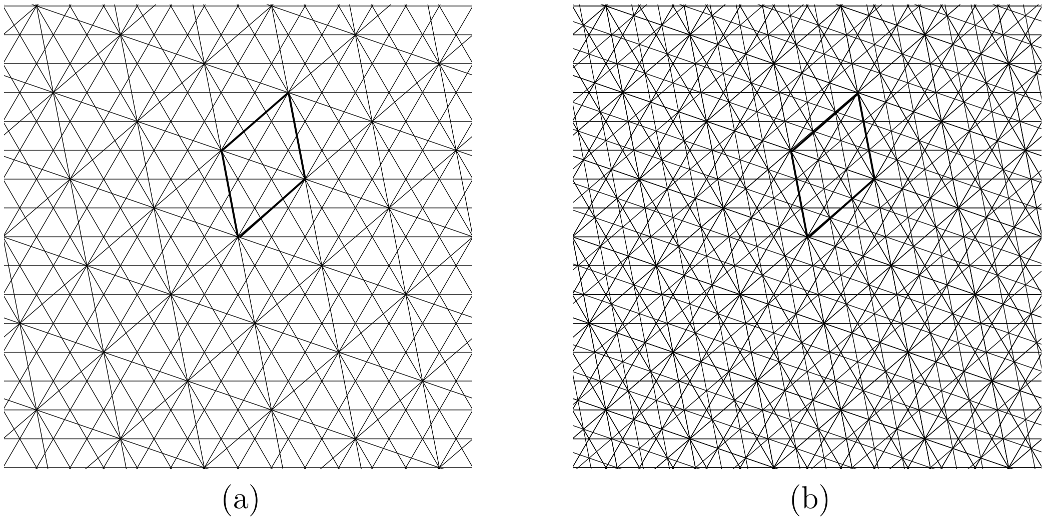


Figure 2.8: Example for $r = 3$: a) Structure of the sublattice $S\Lambda$ with $a = 1, b = 3$; b) Structure of the EVQ and unit cell.

Example 5 *If we take $a = 1$, $b = 3$, the corresponding hexagonal sublattice will have index $|S\Lambda^2/\Lambda^1| = 7$. The corresponding frame of $r = 2$ and stepsizes are given by:*

$$\boldsymbol{F} = \begin{pmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{3\sqrt{3}}{2\sqrt{7}} & \frac{1}{2\sqrt{7}} \\ \frac{\sqrt{3}}{\sqrt{7}} & \frac{-2}{\sqrt{7}} \end{pmatrix} \qquad \begin{array}{l} \Delta_1^1 = \Delta_2^1 = \frac{\sqrt{3}}{2}\Delta \\ \Delta_1^2 = \Delta_2^2 = \frac{1}{3}\frac{\sqrt{21}}{2}\Delta \end{array} \tag{2.37}$$

The values for d_1^2 and d_2^2 that are used in this example are $d_1^2 = d_2^2 = 3$.

Example 6 *An example for $r = 3$ can be easily constructed from Example 5 by adding 2 vectors which are integral combinations of the vectors that compose the frame for $r = 2$. One additional vector is obtained by summing the first 2 vectors in (2.37). Another vector is obtained by summing the third and fourth vectors* *in (2.37). The addition of these* 2 *vectors do not change the sublattice structure (although we are adding more hyperplanes). This can be seen in Fig. 2.8(a). Fig. 2.8(b) shows the final* EVQ *that is obtained.*

It is also possible to construct periodic EVQ's for higher dimensions using redundant families which are not comprised of orthogonal bases, by means of other types of lattices such as those studied in [30] and [45]. Higher dimensional designs having good symmetry properties are studied in detail in Chapter 3 and can also be found in [102].

2.4 Consistent Reconstruction in Periodic Quantizers

In this section, we analyze how to achieve consistency in periodic quantizers under simple reconstruction algorithms (e.g. linear or look-up table).

2.4.1 Consistency under linear reconstruction using the minimal dual frame

Although the results presented in this section hold for any type of frame and any type of linear reconstruction algorithm, the proofs of these results are much clearer and much more intuitive for the case of linear reconstruction using the minimal dual frame and for tight frames composed of a set of r orthogonal bases. We show in Theorem 2 that, given a frame, a necessary condition to have consistency under linear reconstruction is that the scalar quantizers acting on the coefficients are such that the resulting EVQ has a periodic structure. This result follows basically from the fact that when there is no periodicity in the partition defined by an EVQ, the vertices of any two lattices Λ^{j_1} and Λ^{j_2} ($j_1 \neq j_2$) can have arbitrary relative positions, at least in one of the components, which makes it always possible to find linearly inconsistent cells. On the contrary, when there is periodicity, there is only a finite number of relative positions (see Fig. 2.4) and linear consistency is not precluded.

The proof of this result is exactly the same conceptually for any value of the redundancy r and for any dimension N because the crucial point is just the periodicity in the structure regardless of the underlying frame that is used. Since for higher dimensions N and higher redundancies r, the proof becomes much more tedious without adding anything new conceptually, we reduce the proof to the $r = 2$ and $N = 2$ case. However, for completeness, examples will be shown where linear consistency is satisfied for $r > 2$ in $\mathbb{R}^2$.

We need the following Lemma:

Lemma 3 *Let Λ^1 be a rectangular lattice with $\boldsymbol{M}_{\Lambda^1} = diag[\Delta_1^1, \Delta_2^1]$ and Λ^2 another (generic) lattice whose generator matrix is parameterized as:*

$$\boldsymbol{M}_{\Lambda^2} = \begin{pmatrix} \Delta_1^2 & 0 \\ 0 & \Delta_2^2 \end{pmatrix} \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix} \quad where\ \Delta_1^2, \Delta_2^2 \in \mathbb{R}_+,\ \theta \in]0, \frac{\pi}{2}[\tag{2.38}$$

Then, the following equations:

$$\begin{aligned} \Delta_1^2 \cos(\theta) - \Delta_2^2 \sin(\theta) &= q_1 \Delta_1^1 && (2.39) \\ \Delta_1^2 \cos(\theta) + \Delta_2^2 \sin(\theta) &= q_2 \Delta_1^1 && (2.40) \\ \Delta_1^2 \sin(\theta) + \Delta_2^2 \cos(\theta) &= q_3 \Delta_2^1 && (2.41) \\ \Delta_1^2 \sin(\theta) - \Delta_2^2 \cos(\theta) &= q_4 \Delta_2^1 && (2.42) \\ where \quad q_1, q_2, q_3, q_4 &\in \mathbb{Q}(Rational\ numbers) \end{aligned}$$

are all satisfied iff $\boldsymbol{M}_{\Lambda^2} = diag[1/d_1^2, 1/d_2^2]\boldsymbol{M}_{S\Lambda^2}$ where $S\Lambda^2$ is a sublattice of Λ^1, that is, $\boldsymbol{M}_{S\Lambda^2}$ is given as in (2.19), and $d_1^2, d_2^2 \in \mathbb{Z}_+$.

Proof: See Appendix A.6.

The consequence of this Lemma is that, when Λ^2 meets the conditions of the Lemma, the vertices belonging to Λ^2, which can also be written as:

$$\{\boldsymbol{\omega}_i\} = \{k_1(\Delta_1^2 \boldsymbol{\varphi}_1^2 + \Delta_2^2 \boldsymbol{\varphi}_2^2) + k_2(\Delta_1^2 \boldsymbol{\varphi}_1^2 - \Delta_2^2 \boldsymbol{\varphi}_2^2), \quad k_1, k_2 \in \mathbb{Z}\} \tag{2.43}$$

have only a finite number of different (relative) positions within the cells C^{Λ^1} of the quantizer Q^1 (see for example Fig. 2.5(b)). In Theorem 2, we use this fact so that if any of the previous 4 equations (2.39),(2.40),(2.41),(2.42) is not satisfied, we can always find vertices where at least one component can have any arbitrary position within a cell of the quantizer Q^1, and this allows us to find (linearly) inconsistent cells.

Theorem 2 *If the EVQ is a non-periodic quantizer in $\mathbb{R}^2$, then it is always possible to find a linearly inconsistent cell.*

Proof: See Appendix A.7.

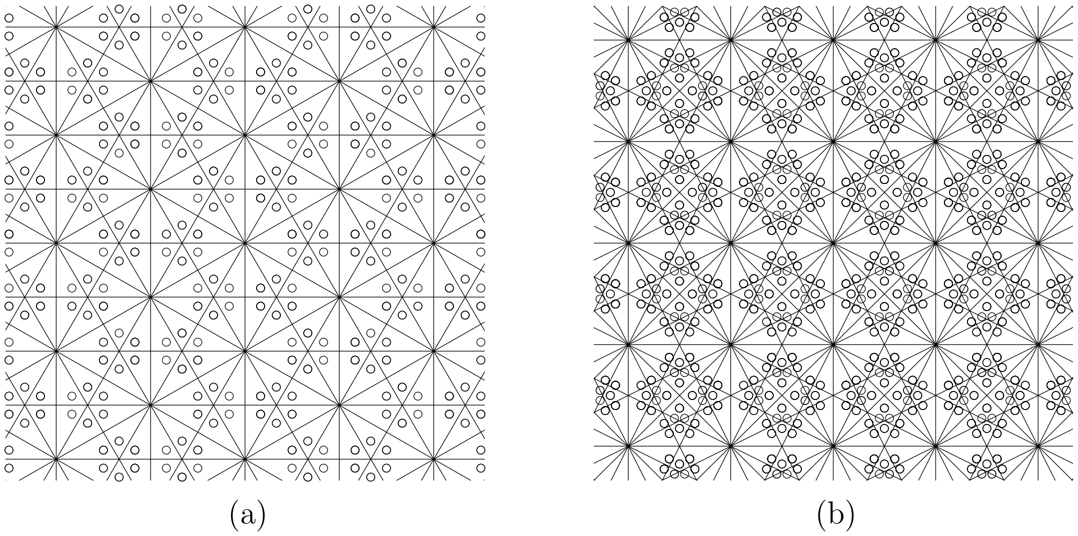


Figure 2.9: Examples of Linearly Consistent Quantizers for a) $r = 3$ and b) for $r = 4$. Minimal dual frame is used for the linear reconstruction.

Thus, periodicity in an EVQ is a necessary condition to achieve consistency under linear reconstruction. Notice that in a periodic EVQ there are only finitely many distinct EVQ cells. Checking whether linear consistency is satisfied, we only need to check on the distinct EVQ cells, which are actually the EVQ cells inside the fundamental polytope of the coincidence site lattice Λ^{CSL}. In fact, given a set of lattices $\Lambda^1, \Lambda^2, \ldots, \Lambda^r$, we can always easily enumerate the positions of the vertices of each of them inside C_o^{CSL} in terms of the corresponding generator matrices and check computationally whether consistency is satisfied or not.

We show in Fig. 2.9 examples of linear consistency in $\mathbb{R}^2$ for redundancies $r = 3, 4$, where the reconstruction vectors have been represented by $'\circ'$.

2.4.2 Consistent reconstruction algorithms with improved performance

Given a regular EVQ, it is desirable for having a good rate-distortion performance that the reconstructions be located near the centroids of the EVQ cells. It can be seen in Fig. 2.9 how the consistent linear reconstructions given by the minimal dual frames for $r = 3, 4$ are not located near the centroids corresponding to a uniform distribution. In order to achieve a better performance, it is necessary to use more intelligent (although simple and low complexity) reconstruction algorithms which make explicit use of the periodicity property.

2.4.2.1 Reconstruction with a small look-up table in Periodic EVQ's

Given a periodic EVQ, it is possible to reconstruct efficiently and accurately by using a small size look-up table scheme, which also ensures consistency. This can be done for any periodic EVQ. Let us first consider the case of tight frames composed by a set of orthogonal bases. Assume, for simplicity and without loss of generality, that $N = 2$ and let $\mathcal{P}_o$ be the smallest rectangular polytope which is a basic unit polytope for the partition defined by the EVQ. Notice that although the minimal unit cell C_o^{CSL} may not be rectangular, from Lemma 1, since Λ^1 is rectangular, it is always possible to find a rectangular polytope $\mathcal{P}_o$ (with volume larger than the volume of C_o^{CSL}) which is also a (non-minimal) basic unit polytope. The reason of choosing this basic rectangular polytope is that the reconstruction algorithm becomes even simpler in this case. Since the periodicity of the EVQ is determined by Λ^{CSL}, the smallest rectangular polytope $\mathcal{P}^{CSL}$ covering C_o^{CSL} is a valid candidate for $\mathcal{P}_o$. It is clear that, due to the periodicity determined by $\mathcal{P}^{CSL}$, any vertical or horizontal shift of $\mathcal{P}^{CSL}$ by an integer number of stepsizes (Δ_1^1 is the horizontal stepsize and Δ_2^1 is the vertical stepsize) gives rise to another polytope which also keeps periodicity.

In Fig. 2.10, the polytope that has been chosen is indicated using bold line. Consider the polytope $\mathcal{P}_o$ and let N_1^1 and N_2^1 be the number of stepsizes that determine the length of the sides (vertical and horizontal) of $\mathcal{P}_o$. For the example

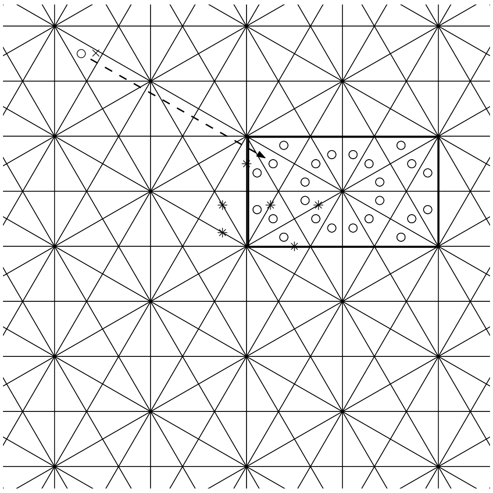


Figure 2.10: Reconstruction algorithm based on look-up table: $'\circ'$ represents reconstruction vectors, $'*'$ the values of the quantized coefficients which define the equivalent cell in the unit cell $\mathcal{P}_o$, 'x' represents the input vector. All the information is first translated to the unit cell $\mathcal{P}_o$, then the reconstruction vector of the equivalent cell is read, and finally it is translated back to the proper cell. Notice that in this example, with this look-up table scheme, the EVQ cells are actually (minimum distance) Voronoi cells.

in Fig. 2.10, $N_1^1 = 2$ and $N_2^1 = 2$. Let $\boldsymbol{v}_o$ be the center of $\mathcal{P}_o$. Given any input signal $\boldsymbol{x}$, it is straightforward to find the equivalent polytope $\mathcal{P}_k$, which is a translation of $\mathcal{P}_o$ given by:

$$\mathcal{P}_k = \mathcal{P}_o + n_1^1 N_1^1 \Delta_1^1 + n_2^1 N_2^1 \Delta_2^1, \quad \text{for some integers} \quad n_1^1, n_2^1 \in \mathbb{Z} \tag{2.44}$$

The basic idea is that given any EVQ cell C_i^{EVQ} it is possible to find very easily and quickly the equivalent cell (by equivalent cell, we mean a congruent cell that is exactly equal in shape and size) which is inside $\mathcal{P}_o$. Given an input signal $\boldsymbol{x}$ whose quantized coefficients are $\boldsymbol{y}_q = PSQ(\boldsymbol{y})$, where $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$, it is possible to translate the values of the quantized coefficients to other values $\boldsymbol{y}_q^{\mathcal{P}_o}$ which define the equivalent cell $V_{i_o}^{EVQ}$ that is inside $\mathcal{P}_o$. This translation is illustrated in Fig. 2.10. Let $\boldsymbol{v}_k$ be the center of the polytope $\mathcal{P}_k$. In this particular case, since $\mathcal{P}_o$ is rectangular (cubic in higher dimensions), it is clear that $\boldsymbol{v}_k$ can be calculated by a simple floor operation because $\mathcal{P}_o$ is rectangular. If we let $\boldsymbol{d} = \boldsymbol{v}_o - \boldsymbol{v}_k$, then if $\hat{\boldsymbol{x}}_o$ is the reconstruction vector corresponding to $V_{i_o}^{EVQ}$, the reconstruction corresponding to C_i^{EVQ} is just $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{\boldsymbol{o}} - d$. The reconstruction $\hat{\boldsymbol{x}}_{\boldsymbol{o}}$ is obtained by just looking up the corresponding reconstruction vector stored in a look-up table. Notice that we can perform optimal reconstruction for the case of a uniform input distribution, because, for each EVQ cell inside $\mathcal{P}_o$, we can store a vector obtained by averaging over all the vertices (extreme points) of the cell (barycenter of the cell), which can be shown to be exactly equal to the centroid of the corresponding (convex) cell assuming a uniform distribution [56]. In the example shown in Fig. 2.10 the needed look-up table consists of only 24 reconstruction vectors. The fundamental advantage provided by the periodicity is that if the periodic EVQ is well designed, the size of the look-up table can be made small, and does not increase with the rate of the EVQ. Notice also that for this example, with the reconstructions given by the look-up table, the EVQ cells are actually (minimum distance) Voronoi cells. For the case of arbitrary EVQ's, a valid polytope $\mathcal{P}_o$ is always given by C_o^{CSL} and a similar reconstruction procedure can be followed. Now, $\boldsymbol{v}_k$ will be calculated by quantizing with respect to Λ^{CSL} which will not be in general a rectangular lattice. For instance, for those periodic EVQ's based on hexagonal lattices in $\mathbb{R}^2$, a valid polytope $\mathcal{P}_o$ will be an hexagonal cell. For instance, in Fig. 2.8, a valid $\mathcal{P}_o$ is illustrated.

Because of the periodicity in the structure of any periodic EVQ, the information can be easily encoded in an embedded (succesive) manner by dividing it into two parts, the entropy associated with the cells $\{\mathcal{P}_k\}$, and the conditional entropy associated with the structure of cells that is inside each $\mathcal{P}_k$, which is the same structure as in $\mathcal{P}_o$. In Fig. 2.10, for instance, given a certain polytope $\mathcal{P}_k$, which can be found by quantizing the coefficients $\{y_1^1, y_2^1\}$ respectively with stepsizes $2\Delta_1^1$ and $2\Delta_2^1$ (this can be viewed as a coarse prequantization), the only additional information that has to be stored to encode a vector is an index between 1 and 24.

The vectors of the look-up table can be easily calculated in any dimension N by using linear programming. In order to do so, for each EVQ cell in the polytope $\mathcal{P}_o$, we run a large enough number of linear programs with different cost vectors pointing in different directions in $\mathbb{R}^N$ and where the constraints are such that they define the specific EVQ cell in terms of inequality constraints. This allows us to calculate all the vertices of the corresponding EVQ cell and by taking the average we obtain a good approximation of its centroid. Moreover, it is not necessary to calculate the vectors of the look-up table for each rate of the EVQ because by linearity, all the vertices scale their coordinates linearly and simmultaneously with Δ_1^1. Therefore, we only need to calculate these vectors *once* for the rate corresponding to $\Delta_1^1 = 1$. This procedure is explained in greater detail in [102] and Chapter 3.

2.4.2.2 Improved linear reconstruction in Periodic EVQ's with spherical symmetry

It is also possible to design periodic EVQ's with additonal symmetry properties so that a very simple improved linear reconstruction algorithm can be used to obtain reconstructions that are located near the centroids of the EVQ cells (assuming a uniform distribution).

Let us consider a periodic EVQ that satisfies the following 2 properties:

1. It is consistent under the usual linear reconstruction using the minimal dual frame.

2. These linear reconstruction vectors are located with circular symmetry (spherical symmetry for $N > 2$) with respect to the lattice points of either the coincidence site lattice Λ^{CSL} or a coset (translation) of it.

Several examples have been found where this circular symmetry is satisfied, as for instance, the two examples shown in Fig. 2.11 for redundancies $r = 2$ and $r = 3$, and the example 4 for dimension $N = 4$ and $r = 3$. The circular symmetry

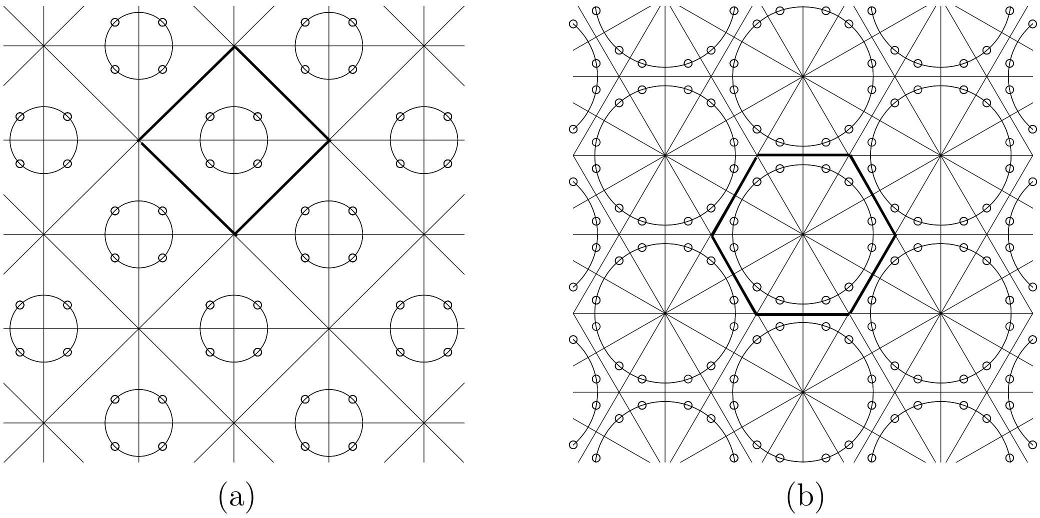


Figure 2.11: Examples of circular symmetry in $\mathbb{R}^2$: a) $r = 2$. Squares represent the lattice points of a coset of Λ^{CSL}; b) $r = 3$. Squares represent the lattice points of Λ^{CSL}.

makes it simple to design a perturbation so that the reconstruction vectors that are obtained are close to the centroids with respect to a uniform distribution.

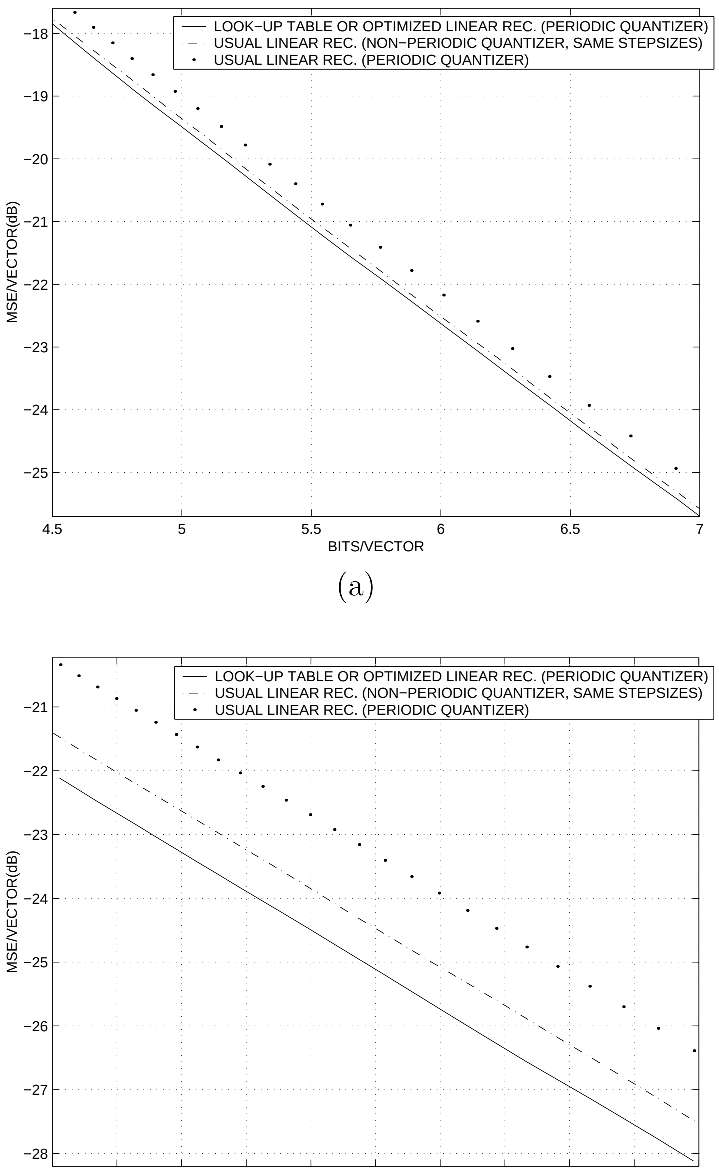
Let $\hat{\boldsymbol{x}}_{LQ}$ be the reconstruction given by a usual lattice quantizer with reproduction vectors given by the points of the coincidence site lattice or a translation of it. For the examples shown in Fig. 2.11, the points of these lattices are represented by squares and one of the Voronoi cells is also highlighted with bold lines. It is very simple to improve the linear reconstruction given by the minimal dual frame by performing a perturbation:

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{MD} + \delta\Delta_1^1(\hat{\boldsymbol{x}}_{MD} - \hat{\boldsymbol{x}}_{LQ}) \tag{2.45}$$

where $\hat{\boldsymbol{x}}_{MD}$ is the reconstruction given by the minimal dual frame and the direction of the perturbation is determined by the difference vector $\boldsymbol{d} = \hat{\boldsymbol{x}}_{MD} - \hat{\boldsymbol{x}}_{LQ}$. Thus, the magnitude of the perturbation is $\|\boldsymbol{d}\|\delta\Delta_1^1$ and the value of δ has to be chosen appropiately so that the final reconstruction $\hat{\boldsymbol{x}}$ is as close as possible to the centroid of the cell. Note that once the best value for δ has been chosen, this is fixed and independent of the input vector $\boldsymbol{x}$ and the scaling of the lattices changes only Δ_1^1. The main advantage of this method with respect to the look-up table scheme is that we do not need a look-up table to store the reproduction vectors of the cells contained inside the minimal periodic unit of the tesselation. However, further research is necessary in order to understand what are the necessary and sufficient conditions which ensure that the property of circular symmetry is satisfied.

2.5 Numerical results for some periodic EVQ designs and Applications

Our designs are more suitable to be used for small redundancies and low to moderate dimensions, and have a complexity similar to the usual linear reconstruction. At high redundancies, it is always possible to find designs but they may not be very efficient in terms of coding due to the number of constraints in the quantization stepsizes that have to be met and also the number of reproductions which



(a)

(b)

Figure 2.12: Comparison, for a 2-dimensional uncorrelated Gaussian source, of (1) usual linear reconstruction with a non-periodic quantizer with equal quantization stepsizes (classic system); (2) reconstruction in a periodic EVQ with different quantization stepsizes using either the look-up table scheme or the improved linear reconstruction (the difference in performance for these two systems is negligible for these examples); (3) usual linear reconstruction in a periodic quantizer with different quantization stepsizes. The values of MSE are given per vector in dB and the bit rate is given in bits/vector; (a) corresponds to the example shown in Fig. 2.11(a) with $r = 2$ and (b) corresponds to the example shown in Fig. 2.6(a) with $r = 3$

have to be stored in the look-up table may be large. However, note that for some important applications such as those involving very high-frequency analog signals (e.g. optical signals), it is usually not feasible to use redundancies higher than $r = 3$ or $r = 4$. Moreover, there exist also other systems called Polyphase A/D converters [91, 51] which divide the bandwidth of the input signal into different narrow subbands (low dimension), and use a different low-rate A/D converter for each of the subband signals, that is, where each of these A/D converters works at a low oversampling ratio. Our system can also be designed theoretically for many different dimensions as shown by Sloane and Beferull-Lozano in [102] and Chapter 3 but the generated tesselations can become very complicated for dimensions $N > 8$ and the number of elements in the look-up table can become also large. For $N \leq 8$, it is possible to find constructions such that the number of different cells (number of elements in the look-up table) is sufficiently small.

We have compared the rate-distortion performance of a) usual linear reconstruction (minimal dual frame) with a non periodic EVQ with equal quantization stepsizes, that is, the quantization system used in all the previous work; b) reconstruction based on a periodic EVQ with different quantization stepsizes using either the look-up table scheme or the improved linear reconstruction (their difference in performance is negligible in these examples) and c) usual linear reconstruction (minimal dual frame) used with a periodic EVQ with different quantization stepsizes. The bit rate associated with the quantized tight frame coefficients is obtained by measuring the joint entropy of all these quantized coefficients, and the distortion is measured in terms of the MSE. The input source that has been used is a 2-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$ with $\sigma = 0.3$. The periodic EVQ's that have been used are the ones shown in Fig. 2.11(a) and Fig. 2.6(a), respectively for $r = 2$ and $r = 3$. For these 2 examples, the rate-distortion performances of the look-up table scheme and the improved linear reconstruction using a periodic EVQ are approximately the same because the reconstructions can be taken to be practically the same and obviously, the associated rate is also the same.

It can be seen in Fig. 2.12 that the best performance is clearly achieved by the look-up table and the improved linear reconstruction systems, with a gain of around 0.2 dBs for $r = 2$ and a gain of around 0.7 dBs for $r = 3$ over the classic system that uses linear reconstruction and the same quantization stepsizes.

At the same time, Fig. 2.12 also shows clearly the fact that, a linearly consistent EVQ does not necessarily yield a better rate-distortion performance than a *different* linearly non-consistent EVQ' at the same rate, that is, by enforcing a periodic structure we may get a quantizer with worse performance than another quantizer whose structure results in linear inconsistency; however, when we use a periodic EVQ and enforce the consistent reconstructions to be sufficiently close to the real centroids by using our reconstruction methods, the periodic EVQ achieves, in all cases, a superior performance over the non-periodic EVQ.

2.5.1 Implications for Oversampled A/D conversion

It can be shown that the oversampling of a periodic bandlimited signal can be expressed as a frame operator in $\mathbb{R}^N$ whose input are the Fourier coefficients (finite discrete Fourier expansion) of the signal that is sampled [84, 111]. As a particular illustrative case, if we consider the space of sinusoids of period T spanned by $\{\cos(2\pi t/T), \sin(2\pi t/T)\}$, the sampling and uniform scalar quantization in amplitude of these signals is equivalent to the quantization of an overcomplete expansion (frame) in $\mathbb{R}^2$. Each sampling time t_i is directly associated with the vector $\boldsymbol{\varphi_i} = [\cos(2\pi t_i/T), \sin(2\pi t_i/T)]$ and all these vectors define the equivalent frame in $\mathbb{R}^2$. Moreover, by Parseval's Theorem, we have that $MSE = \|\hat{y}(t) - y(t)\|_T^2 = \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2$, where $\hat{y}(t)$ is the reconstructed sinusoid, that is, the MSE of the reconstructed sinusoidal signal in the converter is the same

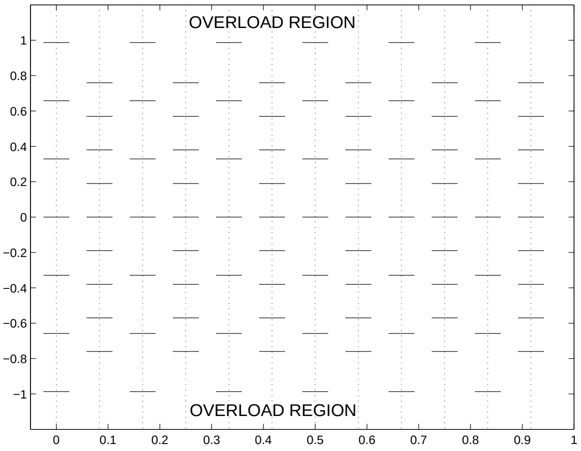


Figure 2.13: Scalar quantizers (time domain) corresponding to the EVQ in Fig. 2.6(a)

as the MSE that occurs on the frame domain. Thus, given a tight frame in $\mathbb{R}^2$ together with a set of different stepsizes such that a periodic EVQ is obtained, if we translate the values of angles to sampling times, we can obtain the scalar quantizers that are applied at the corresponding sampling times. For instance, the quantizer in Fig. 2.6(a) gives rise to a converter with uniform sampling in time and with two different scalar quantizers, one with a stepsize larger than the other one (see Fig. 2.13).

2.6 Conclusions

The basic results presented in this chapter are as follows. We study the problem of achieving consistency in quantized overcomplete expansions with low complexity algorithms. Consistency leads to equivalent vector quantizers which are regular. In order to achieve this goal, we allow the use of different stepsizes in the scalar quantization of the expansion coefficients and construct equivalent vector quantizers (EVQ) having cells with a periodic structure. Periodic quantizers are defined in terms of lattices and sublattices with certain properties and we give various design examples based on different tight frames. On the one hand, we show that periodicity is a necessary condition to have consistency under simple linear reconstruction. On the other hand, a periodic structure makes it possible to reconstruct efficiently and accurately using either a small look-up table whose size does not increase with the rate of the quantizer or using a simple improved linear reconstruction for periodic EVQ's with certain convenient structural properties. Regarding future work, it should be noticed that further research is needed in order to make it possible to apply our approach to A/D conversion of arbitrary band-limited signals.

Chapter 3

Periodic Quantizers based on good Lattice Intersections: Construction and Analysis*

3.1 Introduction and Motivation

In Chapter 2, we have studied the design of the different bases involving the overcomplete expansion together with a set of scalar quantizers so that a periodic tesselation was created. Equivalently, we were finding a set of simple lattices (ideally cubic lattices) $\Lambda^1, \Lambda^2, \ldots, \Lambda^r$ whose intersection was not empty, thus leading to the periodicity of the quantization cells of the EVQ resulting from those lattices.

In this chapter, we pose and solve a different question. Let Λ be an N-dimensional lattice Λ and Q_Λ be the associated lattice vector quantizer (LVQ)

*Some of the work in this chapter was published in [102]. This work was carried out in collaboration with N. J. A. Sloane at AT&T Shannon Laboratory.

based on Λ [34, 55, 62] defined as a mapping $Q_\Lambda : \mathbb{R}^N \rightarrow \mathbb{R}^N$ which maps $\boldsymbol{x} \in \mathbb{R}^N$ to the closest lattice point, i.e., to the lattice point at the center of the Voronoi cell containing $\boldsymbol{x}$ (in the case of a tie, one of the closest lattice points is chosen at random). Suppose now that we decompose Λ as the intersection of N-dimensional lattices $\Lambda^1, \ldots, \Lambda^r$. Then, we can consider replacing the quantizer Q_Λ by the "multiple description quantizer" defined as the product vector quantizer (PVQ):

$$Q_{\Lambda^1} \times Q_{\Lambda^2} \times \ldots \times Q_{\Lambda^r} \ . \tag{3.1}$$

which, given an input vector $\boldsymbol{x}$, simultaneously quantizes it with respect to each of the Λ^j. This gives rise to a different partition of $\mathbb{R}^N$: the new cells are now the intersections of the Voronoi cells of the individual Λ^j, and the r outputs obtained with the (PVQ) in (3.1) specifies uniquely the cell to which $\boldsymbol{x}$ belongs. Given this cell, one reconstructs with the center of that cell using a look-up table scheme. Notice that the tesselation that is created is periodic because of the intersection property. Therefore, these type of quantizers are constructed following these steps: (a) Decompose Λ as the intersection of N-dimensional lattices $\Lambda^1, \ldots, \Lambda^r$, (b) Apply each individual LVQ Q_{Λ^j} giving rise to a lattice tesselation of the space with all the cells being congruent, (c) Find the intersection between the cells of all the individual quantizers, which yields a periodic tesselation, (d) Reconstruct with the center of the cell where the input signal $\boldsymbol{x}$ falls in.

Intuitively, if Q_Λ is a good lattice quantizer, the periodic quantizer $Q_{\Lambda^1} \times Q_{\Lambda^2} \times \ldots \times Q_{\Lambda^r}$ with $\Lambda = \Lambda^1 \cap \Lambda^2 \cap \ldots \cap \Lambda^r$ is expected to yield a tesselation that will contain some cells that are congruent to the Voronoi cell defined by Q_Λ which is a good cell in terms of its quantization performance. Thus, it is legitimate to consider the following problem. Given a good lattice Λ (e.g., such that Q_Λ is a good lattice quantizer), we want to find a set of lattices $\{\Lambda^1, \Lambda^2, \ldots, \Lambda^r\}$ such that the following properties are satisfied:

1. $\Lambda = \Lambda^1 \cap \Lambda^2 \cap \ldots \cap \Lambda^r$.

2. $\Lambda^1, \Lambda^2, \ldots, \Lambda^r$ are as simple as possible. Ideally, we would like these lattices to be cubic.

3. The periodic quantizer $Q_{\Lambda^1} \times Q_{\Lambda^2} \times \ldots \times Q_{\Lambda^r}$ has as good a rate-distortion performance as possible.

Notice that in Chapter 2 we did not start by imposing a certain good intersection lattice Λ such that its corresponding quantizer Q_Λ was good but we went in the other direction. Inspired from this work, we now try to see if forcing the intersection to be a good lattice, this results in good periodic quantizers.

Next, we identify several reasons that motivate the study of these periodic quantizers:

1. If the Λ^j are simpler lattices than Λ, then it may be easier to compute the reconstruction given by the periodic quantizer $Q_{\Lambda^1} \times Q_{\Lambda^2} \times \ldots \times Q_{\Lambda^r}$ than

obtaining the reconstruction given by Q_Λ. Notice however that, in general, for a given input vector $\boldsymbol{x}$, $Q_\Lambda(\boldsymbol{x})$ is different than the reconstruction obtained using the periodic quantizer $Q_{\Lambda^1} \times Q_{\Lambda^2} \times \ldots \times Q_{\Lambda^r}$.

2. This approach could lead to new insights on the so-far intractable problem of finding good lattice quantizers in high dimensions (cf. [34, Chap. 2]). Even in 24 dimensions the best lattice quantizer presently known, the Leech lattice, is very complicated to analyze and to implement — its Voronoi cell has 16969680 faces and over 10^{21} vertices ([34, Chaps. 21, 22, 23, 25], [7], [116], [117]).

3. The individual $Q_{\Lambda^j}(\boldsymbol{x})$ could be communicated over separate channels; in the event of one or more channels failing a reasonably good approximation to $\boldsymbol{x}$ will still be obtained. Other multiple description quantizers have recently been studied in, for example, [45], [115].

4. This approach could lead to quantizers with a lower mean squared error than those obtained from usual lattice quantizers. The 4 particular numerical examples we give in this work do not show favorable results but this work still gives the first basic step towards the solution of this problem and some future work is considered in Section3.5.

5. The study of new decompositions of complicated lattices into simple lattices is also useful in terms of finding simple lattices nested into complicated lattices. Nested lattices have been used lately for a large variety of communication problems when side information is available to either the encoder or the decoder [137].

Good symmetry properties are required because any asymmetry will yield cells with bad shapes and very complicated tesselations of the space which are not useful in practice. There is a slight difference with respect to the setting given in Chapter 2, which is that now each individual quantizer Q_{Λ^j} is a usual LVQ having $\mathbf{0}$ as a reconstruction point while in Chapter 2 the $\mathbf{0}$ vector is not a reconstruction vector but a point belonging to the boundary of the quantizers, as it is usual in the context of A/D conversion related applications. We want to clarify that this is done without loss of generality because in this chapter, our focus is more on the rate-distortion performance of periodic quantizers rather than A/D conversion related applications. All the constructions and designs given in this chapter can be converted to the case where each lattice Λ^j gives the boundary points of the cells of a periodic quantizer by shifting appropiately each individual quantizer Q_{Λ^j}, so that the reconstruction points in each individual quantizer are given now by a shift $\boldsymbol{c}+\Lambda^j$ of the lattice Λ^j. However, notice that obviously this will change the shape of the cells!. By forcing each individual quantizer Q_{Λ^j} to have $\mathbf{0}$ as a reconstruction point and also forcing the intersection to be a good lattice Λ, we expect to get good cells. This is clarified with the following example.

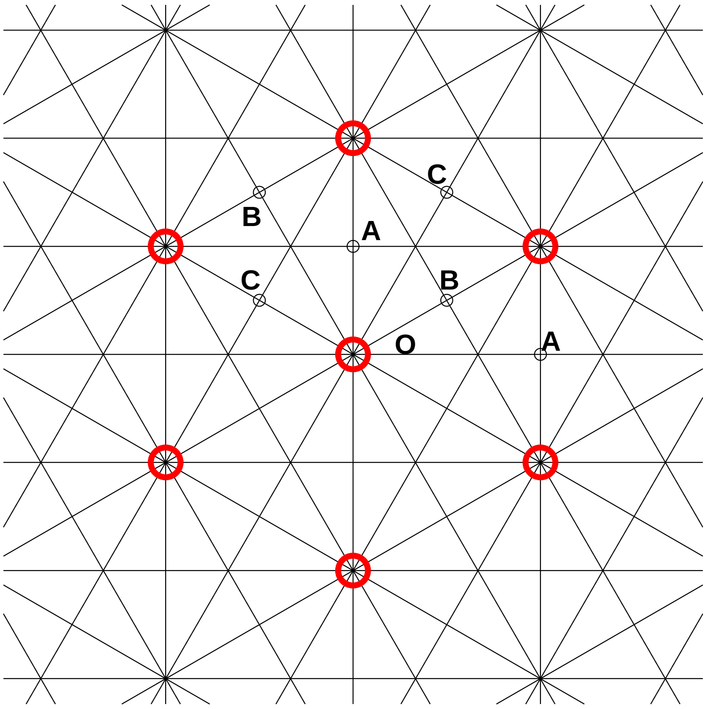


Figure 3.1: The hexagonal lattice (heavy circles) as the intersection of three rectangular lattices (spanned by the vectors OA, OB and OC resp.).

Consider the following appealing example, shown in Fig. 3.1. In Chapter 2, we showed a similar example in Fig. 2.6(a) which is distinct from Fig. 3.1 in that, as just described, the set of reconstruction points does not contain the origin. The familiar planar hexagonal lattice A_2 (large circles) can be obtained as the intersection of three rectangular lattices, all rotations of each other by 120°: these are the lattices generated respectively by the two vectors OA, the two vectors OB and the two vectors OC. If we use generator matrices to specify these lattices (the rows span the lattices) then the three rectangular lattices Λ^1, Λ^2, Λ^3 have generator matrices

$$\begin{pmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{-\sqrt{3}}{2} & \frac{3}{2} \end{pmatrix}, \quad \begin{pmatrix} \frac{-\sqrt{3}}{2} & \frac{1}{2} \\ \frac{\sqrt{3}}{2} & \frac{3}{2} \end{pmatrix} \tag{3.2}$$

and their intersection has generator matrix

$$\begin{pmatrix} \sqrt{3} & 1 \\ 0 & 2 \end{pmatrix}, \tag{3.3}$$

which is indeed a copy of the A_2 lattice. In this case, the tesselation contains four kinds of cells, as shown by the heavy (solid) lines in Fig. 3.2. For example, if the point $\boldsymbol{x}$ being quantized is close to $[0,0]$, then $Q_{\Lambda^1}(\boldsymbol{x}) = Q_{\Lambda^2}(\boldsymbol{x}) = Q_{\Lambda^3}(\boldsymbol{x}) = [0,0]$, and the cell containing $\boldsymbol{x}$ is the intersection of the Voronoi cells containing $[0,0]$ of the three Λ^j. This is the horizontally shaded hexagon in Fig. 3.2. Just to

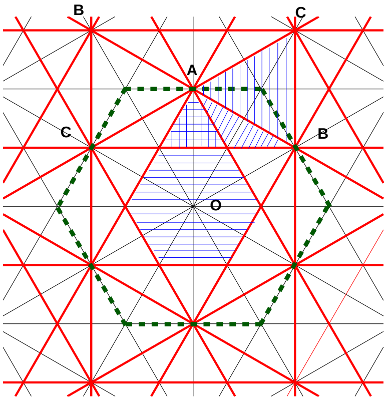


Figure 3.2: Tesselation associated with the cells in Fig. 3.1.

the North of this hexagon the cell is the intersection of the Voronoi cell for Λ^1 that contains $[0, 1]$ with the Voronoi cells at $[0, 0]$ for Λ^2 and Λ^3: this is the small cross-hatched equilateral triangle. There are two further cells that are obtained in a similar manner: the diagonally shaded isosceles triangle and the larger vertically shaded equilateral triangle. Notice that if we had used the individual quantizers as in Chapter 2, the resulting cells would have been the triangles that can be seen in Fig. 3.1. In that case, if $\boldsymbol{M}_{\Lambda^j}$ is the generator matrix of Λ^j, the reconstruction points for each individual quantizer are given by $[k_1 + 1/2, k_2 + 1/2]\boldsymbol{M}_{\Lambda^j}$, the Voronoi cells for each individual quantizer whould have boundary points given by Λ^j and the intersection of these cells would be the triangles in Fig. 3.1. On the other hand, in this chapter, by making each individual quantizer Q_{Λ^j} have $[0, 0]$ as a reconstruction point, when performing the intersection of the Voronoi cells for the different individual quantizers, we obtain an hexagonal cell in Fig. 3.2 for the periodic quantizer. In summary, the lattice decomposition of Λ as an intersection of the individual lattices $\Lambda^1, \ldots, \Lambda^j$ is kept the same, but the individual quantizers that are used are related by shift, which implies also that the Voronoi cells are related by the same shift, which eventually, when calculating the intersection of the Voronoi cells of the individual quantizers, results in a different tesselation in each case for the periodic quantizer. Keeping this in mind, the main issue here is to obtain the lattice decompositions. The tesselation will be implied directy by this decomposition and the choice we have taken in this chapter for the individual quantizers Q_{Λ^j}, $j = 1, \ldots, r$.

The present work was also in part prompted by the question of how to find good constructions of periodic quantizers for dimensions higher than 2, since, as it is well known, the performance in quantization increases as the dimension is increased.

In Section 3.2, we introduce some terminology and notation. Section 3.3 describes some general constructions for writing a lattice as an intersection of a small number of simpler, decomposable lattices in different dimensions. We give a number of examples, including the body-centered cubic (bcc) and face-centered cubic (fcc) lattices D_3^* and D_3, the root lattices D_4, E_6^*, E_8, the Coxeter-Todd lattice K_{12}, the Barnes-Wall lattices BW_n and the Leech lattice Λ_{24}. We focused attention on these lattices because A_2, D_3^*, D_4, E_6^*, E_8, K_{12}, BW_{16} and Λ_{24} are the best quantizers currently known[2] in their dimensions [34]. In fact, A_2 is optimal among all two-dimensional quantizers [113], and D_3^* is optimal among three-dimensional lattice quantizers [6].

Table 3.1 summarizes the main decompositions mentioned in this work. The resulting periodic tesselations or honeycombs[3] have not been studied before.

[2] Assuming always that the random vector to be quantized is uniformly distributed over a large ball in $\mathbb{R}^N$.

[3] Of course there is an extensive literature dealing with the Voronoi and Delaunay cell decompositions associated with various lattices — see for example [31], [35], [33], [34], [37].

Lattice	Copies	Component lattice	Sections
A_2	3	rectangular	$3.1, 3.3, 3.4.1$
A_3^* (bcc)	3	"rectangular"	$3.3, 3.4.2$
A_3 (fcc)	4	"prismatic"	$3.3, 3.4.3$
D_4	3	$(\mathbb{Z})^4$	$3.3, 3.4.4$
E_6^*	4	$(A_2^*)^3$	3.3
E_8	15	$(\mathbb{Z})^8$	3.3
E_8	10	A_2^4	3.3
E_8	5	D_4^2	3.3
K_{12}	21	A_2^6	3.3
Leech	4095	$(\mathbb{Z})^{24}$	3.3
BW_N	$\prod_{j=1}^{m-1}(2^j - 1)$	$(\mathbb{Z})^N$	3.3

Table 3.1: Summary of decompositions described in this chapter.

In Section 3.4 we first analyze the geometry of the tesselations (decompositions of space into cells) associated with the periodic quantizers is analyzed and we give a general expression for the normalized (dimensionless) second moment for any periodic quantizer at high rates. Then, we determine the honeycombs (periodic tesselations) and mean squared errors for the cases of the A_2, bcc, fcc and D_4 lattices, respectively. The final Section3.5 contains some conclusions and comments.

3.2 Notation

Let Λ be a lattice in $\mathbb{R}^N$. The dual lattice will be denoted, as in Chapter 2, by Λ^*. The *norm* of a vector $\boldsymbol{x} \in \mathbb{R}^N$ in this chapter is taken as its squared length $\langle \boldsymbol{x}, \boldsymbol{x} \rangle$. A *similarity* σ is a linear map from $\mathbb{R}^N$ to $\mathbb{R}^N$ such that there is a real number

n with $\langle \sigma(\boldsymbol{x}), \sigma(\boldsymbol{y}) \rangle = n\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. If Λ and M are similar lattices we write $\Lambda \cong M$. A lattice Λ is said to be *n-modular* if $\Lambda \cong \Lambda^*$ under a similarity that multiplies norms by n. For example, the root lattices $(\mathbb{Z})^N$ and E_8 are 1-modular, A_2 is 3-modular, and D_4 is 2-modular. We write $\Lambda^1 + \Lambda^2 + \ldots + \Lambda^r$ for the lattice generated by the basis vectors of thelattices $\Lambda^1, \Lambda^2, \ldots$. Two lattices or polytopes are *congruent* if one can be mapped to the other by an element of the special orthogonal group $SO(N)$, which is the group of isometries (distance-preserving transformations of the space) in $\mathbb{R}^N$. $Aut(\Lambda)$ represents the automorphism group of a lattice, that is, the set of isometries ($SO(N)$) that fix the origin and take the lattice to itself[4]. S_j will denote the symmetric group, that is, the set of all the permutations of j elements ($j!$ possible permutations). A lattice Λ_i is said to be decomposable if it is the direct sum of congruent copies of one or more lattices with lower dimension. For instance, $\Lambda_i = (\mathbb{Z})^2$ is the direct sum of 2 congruent copies (related by a rotation of 90° degrees) of the one-dimensional lattice $\mathbb{Z}$.

3.3 Writing a lattice as an intersection

Let Λ be a lattice in $\mathbb{R}^N$. We wish to write Λ as an intersection

$$\Lambda = \Lambda^1 \cap \Lambda^2 \cap \ldots \cap \Lambda^r \tag{3.4}$$

[4] The automorphism of the hexagonal lattice for instance has order 12 since it is generated by a rotation through 60° and a reflection in a line joining the center of two spheres

where r is small and the Λ^j are pairwise congruent and as "simple" as possible. Ideally we would like each Λ^j to be a direct sum of congruent copies of a fixed low-dimensional lattice K such as $\mathbb{Z}$, A_2 or D_4, but this is not always possible. In the example shown in Fig. 3.1, for instance, the lattices Λ^j's are rectangular rather than square lattices. If this is not possible, we ask that the Λ^j be decomposable into a direct sum of as many congruent low-dimensional sublattices as possible.

If we were going to investigate the honeycombs associated with higher dimensional intersections such as those for E_6^* or E_8, we would impose an additional formal requirement that the Λ^j form an orbit under some subgroup of the automorphism group $Aut(\Lambda)$, in order to guarantee that the honeycomb be symmetric. However, for the low-dimensional cases, we have been able to achieve this symmetry by using the natural decompositions, without introducing the machinery of group theory.

Summarizing, these are the requirements imposed for the decomposition:

- $\{\Lambda^j\}_{j=1}^r$ are pairwise congruent, that is, it is possible to go from one to the other through a rotation and/or a traslation. This will result in good symmetry properties.

- Each Λ^j is as simple as possible, but not necessarily cubic. This will result in fast decoding with respect to the cells of the resulting tesselation.

- Each Λ^j is decomposable in one of the following ways:

1. It is a cartesian product of a lattice fixed lattice Λ_o, that is, $\Lambda^j = K \times \ldots K = (K)^m$, where K is a good low dimensional lattice such as $\mathbb{Z}, A_2, D_4$.
2. Cartesian product of different simple low-dimensional lattices.

This Section describes some general methods for finding intersections. We will make use again as in Chapter 2 of the de Morgan's law [74, 87]. Since it is crucial for finding the different constructions, we rewrite it again for clarity and completeness:

Theorem 3 *If* $\Lambda^1, \ldots, \Lambda^r$ *are lattices in* $\mathbb{R}^N$ *then*

$$\Lambda^1 \cap \ldots \cap \Lambda^r \;=\; ((\Lambda^1)^* + \ldots + (\Lambda^r)^*)^* \ .$$

Method 1: Partitioning the minimal vectors for Λ^*. The first method is given by the following proposition:

Proposition 1 *Given an* N*-dimensional lattice* Λ*, if:*

1. Λ^* *is generated by its minimal vectors, i.e. the vectors of minimal nonzero norm.*
2. *Minimal vectors of* Λ^* *can be partitioned into* r *congruent copies (sets) of minimal vectors. Let* $(K)^{N/\kappa}$ *be the lattice generated by one of these sets, where* K *is a lattice of dimension* κ*.*

3. K is also modular, that is, K and K^* are related by scaling, rotation and/or reflection.

then, taking Λ^j to be the j-th copy of the lattice $(K^*)^{N/\kappa}$, the intersection property is satisfied.

Proof: See Appendix B.1

Whether this partitioning is possible is an interesting question in its own right. For example, can the 240 minimal vectors of E_8 be partitioned into 15 copies of the minimal vectors of (a scaled version of) $(\mathbb{Z})^8$, i.e. into 15 coordinate frames[5] or into 10 copies of the minimal vectors of $(A_2)^4$? Partial answers are given below. We have studied constructions for the cases where K is one of $\mathbb{Z}$, A_2 or D_4.

We now give a number of examples, beginning with the case when $K = \mathbb{Z}$.

D_4. The lattice D_4 may be taken to have generator matrix

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{3.5}$$

[5] A *coordinate frame* in $\mathbb{R}^N$ is a set of $2N$ vectors $\pm\boldsymbol{v}_1, \ldots, \pm\boldsymbol{v}_N$ with $\langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle =$ a constant, $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0$ if $i \neq j$.

and then the 24 minimal vectors consist of eight of the form $[\pm 2, 0, 0, 0]$ and 16 of the form $[\pm 1, \pm 1, \pm 1, \pm 1]$. D_4 is 2-modular and is generated by its minimal vectors. The minimal vectors may be partitioned into three coordinate frames, consisting of ± 1 times the rows of each of the matrices

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad \begin{pmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & +1 & -1 & -1 \end{pmatrix}, \quad \begin{pmatrix} -1 & +1 & +1 & +1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \\ -1 & +1 & -1 & -1 \end{pmatrix}. \tag{3.6}$$

After applying the Proposition 1 and rescaling, we conclude that if Λ^1, Λ^2, Λ^3 ($\cong (\mathbb{Z})^4$) have the generator matrices given in (3.6), then

$$\Lambda^1 \cap \Lambda^2 \cap \Lambda^3 = \Lambda\,, \tag{3.7}$$

where Λ has generator matrix

$$\begin{pmatrix} 4 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}, \tag{3.8}$$

and is another version of D_4 on the scale at which its minimal norm is 8. The group generated by the second matrix in (3.6) and diag$[-1, +1, +1, +1]$ is a symmetric group S_3 permuting the Λ^j; it is also a subgroup of $Aut(\Lambda)$.

Barnes-Wall lattices. The preceding example can be generalized using orthogonal spreads. Let BW_N ($N = 2^m$, $m = 1, 2, \ldots$) denote the N-dimensional Barnes-Wall lattice ([34], [81], [82]). In particular, $BW_2 \cong (\mathbb{Z})^2$, $BW_4 \cong D_4$, $BW_8 \cong E_8$. For $N \neq 8$, BW_N is 2-modular, while as already mentioned BW_8 is 1-modular.

It is known that the minimal vectors of BW_N may be partitioned into $\prod_{j=1}^{m-1}(2^j - 1)$ coordinate frames[6], which are transitively permuted by symmetries of $Aut(BW_N)$.

It follows that BW_N can be written as the intersection of $\prod_{j=1}^{m-1}(2^j - 1)$ copies of $(\mathbb{Z})^N$. In particular, E_8 is the intersection of 15 copies of $(\mathbb{Z})^8$. An explicit method for constructing such an intersection for E_8 is given below.

Intersections of smaller numbers of lattices are possible, although they are less symmetric and therefore less satisfactory. For example in (3.7) it is also true that $\Lambda^1 \cap \Lambda^2 = \Lambda \cong D_4$. Similarly, E_8 is (up to a similarity) the intersection of $2(\mathbb{Z})^8$ and the lattice (similar to $2(\mathbb{Z})^8$) with generator matrix:

[6] This is a consequence of the existence of an orthogonal spread in the orthogonal vector space $\Omega^+(2m, 2)$ of maximal Witt index ([21], [22], [81]).

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 & 0 & -1 \end{pmatrix}$$

But this representation of E_8 is in no way canonical, and the resulting honeycomb does not have interesting properties.

Eisenstein and Hurwitzian lattices. Smaller intersections which *are* canonical can be obtained if we change K from $\mathbb{Z}$ to A_2 or D_4. For example, the minimal vectors of E_8 can be partitioned into 10 copies of the minimal vectors of $(A_2)^4$. As in [34], let $\mathcal{E} = \{a + b\omega : a, b \in \mathbb{Z}\}$, $\omega = e^{2\pi i/3}$, denote the ring of Eisenstein integers. The six units in $\mathcal{E}$ are ± 1, $\pm\omega$, $\pm\bar{\omega}$. When regarded as a two-dimensional real lattice $\mathcal{E}$ is similar to A_2. As an $\mathcal{E}$-module[7], E_8 has generator matrix

$$\begin{pmatrix} \theta & 0 & 0 & 0 \\ 0 & \theta & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \end{pmatrix}, \quad \text{where} \quad \theta = \omega - \bar{\omega}\,. \tag{3.9}$$

Inner products are computed using the Hermitian inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum u_i \bar{v}_i$. See [34, Chapters 2 and 7] for further details. The minimal vectors consist of 24 of

[7] For definitions of modules and free modules, see for instance [124]

the form $[u\theta, 0, 0, 0]$, where u is a unit in $\mathcal{E}$, and $8 \times 3^3 = 216$ which are congruent mod θ to one of the eight nonzero codewords of the tetracode [34, Chap. 3].

A partition of these 240 vectors into 10 copies of the minimal vectors of $(A_2)^4$ was found by graph coloring. A graph was constructed with the 40 projectively distinct[8] vectors as nodes and with edges corresponding to pairs of non-orthogonal vectors. A coloring with 10 colors was then found with the help of a program.The ten copies of $\mathcal{E}^4 \cong (A_2)^4$ are shown in Table 3.2. Only one from each complex conjugate pair is shown. By applying the Lemma we obtain a representation of E_8 as an intersection of 10 copies of $(A_2)^4$. In fact (since $\mathcal{E}$ is itself 3-modular) we may omit the final step of taking the duals of the lattices in Table 3.2. Let $\Lambda^1, \ldots, \Lambda^{10}$ be the ten versions of $(A_2)^4$ generated by the rows of Table 3.2 and their complex conjugates. Then their intersection is easily seen to be the version of E_8 with generator matrix θ times (3.9). This decomposition is probably not

θ	0	0	0	0	θ	0	0	0	0	θ	0	0	0	0	θ
0	1	-1	1	1	0	-1	-1	1	-1	0	1	1	1	1	0
0	1	-1	ω	1	0	$-\omega$	$-\bar{\omega}$	1	$-\omega$	0	$\bar{\omega}$	1	ω	ω	0
0	1	$-\omega$	ω	1	0	$-\omega$	$-\omega$	1	-1	0	ω	1	1	ω	0
0	1	$-\omega$	$\bar{\omega}$	1	0	$-\bar{\omega}$	-1	1	$-\omega$	0	1	1	ω	$\bar{\omega}$	0
0	1	$-\omega$	1	1	0	-1	$-\bar{\omega}$	1	$-\bar{\omega}$	0	$\bar{\omega}$	1	$\bar{\omega}$	1	0

Table 3.2: Decomposition of minimal vectors of E_8 into ten copies of $\mathcal{E}^4 \cong (A_2)^4$. Each row generates a copy of $\mathcal{E}^4 \cong (A_2)^4$. The complex conjugates of the last four rows have been omitted.

[8] Two vectors are projectively the same if one is an scalar times the other

unique, and it would be nice to know which version has the largest symmetry group.

Again just two lattices also suffice: E_8 is also the intersection of the first two lattices in Table 3.2.

We may also write E_8 as the intersection of five copies of $(D_4)^2$. For this we regard E_8 as a 2-dimensional module over the ring $\mathbb{H} \cong D_4$ of Hurwitzian quaternions [34, p. 55]. The five copies of $(D_4)^2$ have generator matrices

$$\begin{pmatrix} 1+i & 0 \\ 0 & 1+i \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}, \begin{pmatrix} 1 & j \\ 1 & -j \end{pmatrix}, \begin{pmatrix} 1 & k \\ 1 & -k \end{pmatrix}. \tag{3.10}$$

E_6^* may be written as the intersection of four copies of $(A_2)^3$, with generator matrices

$$\begin{pmatrix} \theta & 0 & 0 \\ 0 & \theta & 0 \\ 0 & 0 & \theta \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \bar{\omega} \\ 1 & \bar{\omega} & \omega \end{pmatrix}, \begin{pmatrix} 1 & 1 & \omega \\ 1 & \omega & 1 \\ 1 & \bar{\omega} & \bar{\omega} \end{pmatrix}, \begin{pmatrix} 1 & 1 & \bar{\omega} \\ 1 & \omega & \omega \\ 1 & \bar{\omega} & 1 \end{pmatrix}. \tag{3.11}$$

This was found by partitioning the 72 minimal vectors of E_6 (by hand) into four copies of the minimal vectors of A_2^3 and using the proposition 1.

Method 2: Congruence bases and norm-doubling maps. The second method is based on the observation that several well-known lattices Λ have the property that for some prime π, the vectors in some of the classes of $\Lambda/\pi\Lambda$ can be partitioned into coordinate frames. For example, Conway's proof of the uniqueness of the Leech lattice Λ_{24} [34, Chap. 12] considers the classes of $\Lambda_{24}/2\Lambda_{24}$. A consequence of the numerical identity

$$\begin{aligned} \frac{n_0}{1}+\frac{n_4}{2}+\frac{n_6}{2}+\frac{n_8}{48} &= \frac{1}{1}+\frac{196560}{2}+\frac{16773120}{2}+\frac{398034000}{48} \\ &= 16777216 = 2^{24}\,, \end{aligned} \tag{3.12}$$

where n_j is the number of vectors in Λ_{24} of norm j, is that, for the classes of $\Lambda_{24}/2\Lambda_{24}$ in which the minimal norm is 8, the minimal vectors in the class form a coordinate frame or *congruence base*. A similar property holds for the D_4, E_8, K_{12} and other lattices (cf. [32]). This gives a representation of Λ_{24} as an intersection of $398034000/48 = 8292375$ copies of $(\mathbb{Z})^{24}$. However, the following argument, due to J. H. Conway (personal communication), shows that if the lattice has a suitable norm-doubling map (cf. [34, p. 239], [32]) then we can also partition the minimal vectors into coordinate frames and obtain a smaller intersection.

Suppose that a lattice $\Lambda \subseteq \mathbb{R}^N$ has the structure of a free module over a ring J with inner product $\langle\ ,\ \rangle$ (cf. [34, Chap. 2]). In the present application J will be either $\mathbb{Z}$ or $\mathcal{E}$. Let $a = N/\dim_{\mathbb{R}} J$[9]. Consider the classes of $\Lambda/2\Lambda$. Suppose there is an integer m with the property that each class either contains no vectors

[9] $\dim_{\mathbb{R}} J$ is the dimension of J as a vector space over the reals

of norm $2m$, or else all the vectors of norm $2m$ in the class can be partitioned into sets of $2a$ vectors $\pm\boldsymbol{v}_1, \pm\boldsymbol{v}_2, \ldots, \pm\boldsymbol{v}_a$ where $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0$ if $i \neq j$.

Suppose in addition there is a *norm-doubling* map $\boldsymbol{T}$, a similarity from Λ into Λ such that $\langle \boldsymbol{Tu}, \boldsymbol{Tu} \rangle = 2\langle \boldsymbol{u}, \boldsymbol{u} \rangle$ for $\boldsymbol{u} \in \Lambda$, with the extra property that $2\Lambda \subset \boldsymbol{T}\Lambda$. Then we may conclude that the vectors of norm m in Λ may also be partitioned into sets of $2a$ mutually orthogonal vectors.

To see this, let $\boldsymbol{u} \in \Lambda$ have norm m. Then $\boldsymbol{v} = \boldsymbol{Tu}$ has norm $2m$, and by the hypotheses is part of a coordinate frame $\pm\boldsymbol{v}_1, \ldots \pm \boldsymbol{v}_a$, where $\boldsymbol{v}_i = \boldsymbol{v} + 2\boldsymbol{w}_i$, say, with $\boldsymbol{w}_1 = \boldsymbol{0}$. We can write $2\boldsymbol{w}_i = \boldsymbol{Tw}'_i$ for some $\boldsymbol{w}'_i$, so $\boldsymbol{v}_i = \boldsymbol{T}(\boldsymbol{u} + \boldsymbol{w}'_i)$. Since $\boldsymbol{T}$ is a similarity, the set $\pm(\boldsymbol{u} + \boldsymbol{w}'_i)$ is a coordinate frame containing $\boldsymbol{u}$.

Examples

(i) $\Lambda = D_4$ or E_8, $J = \mathbb{Z}$, $m = 2$, $\boldsymbol{T} =$ direct sum of 2 or 4 copies of $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, with $\boldsymbol{T}^2 = 2\boldsymbol{I}$. The classes of $D_4/2D_4$ and $E_8/2E_8$ and the associated congruence bases are given in [34, Chap. 6]. The analogue of (3.12) for E_8 reads

$$1 + \frac{240}{2} + \frac{2160}{16} = 2^8 \ .$$

We obtain decompositions of the minimal vectors of D_4 into three coordinate frames, as already seen in (3.6), and of the minimal vectors of E_8 into 15 coordinate frames as also discussed above. To get an explicit decomposition in the latter case, note that a coordinate frame of norm 4 vectors has the form $\boldsymbol{T}\boldsymbol{u} + 2\boldsymbol{w}_i = \boldsymbol{T}(\boldsymbol{u} + \boldsymbol{T}\boldsymbol{w}_i)$. So the coordinate frame of norm 2 vectors consists of the vectors of minimal norm in the translate $\boldsymbol{u} + \boldsymbol{T}\Lambda$. For E_8 these consist of seven sets of the form shown on the left in (3.13) and eight of the form shown on the right:

$$\begin{pmatrix} + & 0 & + & 0 & 0 & 0 & 0 & 0 \\ + & 0 & - & 0 & 0 & 0 & 0 & 0 \\ 0 & + & 0 & + & 0 & 0 & 0 & 0 \\ 0 & + & 0 & - & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & + & 0 & + & 0 \\ 0 & 0 & 0 & 0 & + & 0 & - & 0 \\ 0 & 0 & 0 & 0 & 0 & + & 0 & + \\ 0 & 0 & 0 & 0 & 0 & + & 0 & - \end{pmatrix}, \quad \frac{1}{2}\begin{pmatrix} + & + & + & + & + & + & + & + \\ + & + & + & + & - & - & - & - \\ + & + & - & - & + & + & - & - \\ + & + & - & - & - & - & + & + \\ + & - & + & - & + & - & + & - \\ + & - & + & - & - & + & - & + \\ + & - & - & + & + & - & - & + \\ + & - & - & + & - & + & + & - \end{pmatrix}. \tag{3.13}$$

Then E_8 is also the intersection of the 15 copies of $(\mathbb{Z})^8$ having these generator matrices.

(ii) $\Lambda =$ Leech lattice Λ_{24}, $J = \mathbb{Z}$, $m = 4$, $\boldsymbol{T} = (\boldsymbol{I} + \boldsymbol{i})$, where $\boldsymbol{i} \in Aut(\Lambda_{24})$ is given in [34, Fig. 6.7], and satisfies $\boldsymbol{i}^2 = -\boldsymbol{I}$, with $(\boldsymbol{I}+\boldsymbol{i})(\boldsymbol{I}-\boldsymbol{i}) = \boldsymbol{T}(\boldsymbol{I}-\boldsymbol{i}) = 2\boldsymbol{I}$. The coordinates frames of norm 4 vectors consist of the 48 vectors of minimal norm in the translates $\boldsymbol{u}+(\boldsymbol{I}-\boldsymbol{i})\Lambda$ where $\langle \boldsymbol{u}, \boldsymbol{u}\rangle = 4$. We obtain a decomposition of the minimal vectors of Λ_{24} into 4095 coordinate frames, and a representation of Λ_{24} as the intersection of 4095 copies of $(\mathbb{Z})^{24}$.

We do not know if it is possible to write the Leech lattice as the intersection of 2730 copies of $(A_2)^{12}$. Since 196560/96 is not an integer, there is no analogous decomposition as an intersection of copies of $(D_4)^6$.

(iii) Λ = Coxeter-Todd lattice K_{12}, a 6-dimensional $\mathcal{E}$-module, $J = \mathcal{E}$, $m = 6$, $\boldsymbol{T}$ = the map given in [32, Eq. (43)], with $\boldsymbol{T}^2 + \boldsymbol{T} + 2 = \boldsymbol{0}$. The classes of $K_{12}/2K_{12}$ are given in [32], and the analogue of (3.12) reads

$$1 + \frac{756}{2} + \frac{4032}{1} + \frac{20412}{12} = 2^{12} \ .$$

The coordinate frames of norm 6 vectors consist of the 12 minimal vectors in the translates

$\boldsymbol{u} + (\boldsymbol{T} + \boldsymbol{I})K_{12}$, $\langle \boldsymbol{u}, \boldsymbol{u} \rangle = 6$. By combining these coordinate frames in sets of three, by taking the union of the sets $\alpha\{\pm\boldsymbol{v}_1, \ldots, \pm\boldsymbol{v}_{12}\}$ with $\alpha = 1, \omega$ and $\bar{\omega}$, we obtain a decomposition of the minimal vectors of K_{12} into 21 copies of the minimal vectors of $(A_2)^6$, and, via the Lemma 3, a representation of K_{12} as the intersection of 21 copies of $(A_2)^6$. An explicit decomposition, not shown here, was found by the graph coloring method mentioned earlier.

Suppose in addition that there is a *norm-doubling* map $\boldsymbol{T}$, a similarity from Λ into Λ such that $(\boldsymbol{T}\boldsymbol{u}, \boldsymbol{T}\boldsymbol{u}) = 2\langle \boldsymbol{u}, \boldsymbol{u} \rangle$ for $\boldsymbol{u} \in \Lambda$, with the extra property that

$2\Lambda \subset \boldsymbol{T}\Lambda$. Then we may conclude that the vectors of norm m in Λ may also be partitioned into sets of $2a$ mutually orthogonal vectors. To see this, let $\boldsymbol{u} \in \Lambda$ have norm m. Then $\boldsymbol{v} = \boldsymbol{T}\boldsymbol{u}$ has norm $2m$, and by the hypotheses is part of a coordinate frame $\pm\boldsymbol{v}_1, \ldots \pm \boldsymbol{v}_a$, where $\boldsymbol{v}_i = \boldsymbol{v} + 2\boldsymbol{w}_i$, say, with $\boldsymbol{w}_1 = 0$. We can write $2\boldsymbol{w}_i = \boldsymbol{T}\boldsymbol{w}_i'$ for some w_i', so $\boldsymbol{v}_i = \boldsymbol{T}(\boldsymbol{u} + \boldsymbol{w}_i')$. Since $\boldsymbol{T}$ is a similarity, the set $\pm(\boldsymbol{u} + \boldsymbol{w}_i')$ is a coordinate frame containing $\boldsymbol{u}$.

Method 3: First principles. If the above methods fail, as they do for the bcc and fcc lattices, we can always fall back on a direct attack from first principles. The following method handles the hexagonal, bcc and fcc lattices in a unified manner. We list the vectors of small norms in the lattice, and look for a partition of some subset of these vectors which produces a small number of congruent, decomposable lattices whose intersection is similar to the original lattice.

For the hexagonal lattice, which we take to be generated by $[0, 1]$ and $[-\frac{\sqrt{3}}{2}, \frac{1}{2}]$, there are six vectors of norm 1, namely $[0, \pm 1]$, $[\pm\frac{\sqrt{3}}{2}, \pm\frac{1}{2}]$, and six of norm 3, namely $[\pm\sqrt{3}, 0]$, $[\pm\frac{\sqrt{3}}{2}, \pm\frac{3}{2}]$. Then (3.2) is obtained by partitioning these 12 vectors into three sets of size 4.

For the bcc lattice generated by $[1, 0, 0]$, $[0, 1, 0]$, $[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$, the vectors of small norms are the following:

shape	norm	number
0 0 0	0	1
$\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	$\frac{3}{4}$	8
1 0 0	1	6
1 1 0	2	12

We take the 18 vectors of norms 1 and 2 and partition them into three sets of size 6. The resulting lattices have generator matrices

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix} \tag{3.14}$$

and their intersection has generator matrix

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \tag{3.15}$$

which is indeed another version of the bcc lattice[10]. This decomposition of the bcc lattice is the simplest we have found.

For the fcc lattice generated by $[1,1,0]$, $[1,-1,0]$, $[0,1,-1]$, the vectors of small norms are:

[10] Strictly speaking, we obtain $2D_3^*$

shape	norm	number
0 0 0	0	1
1 1 0	2	12
2 0 0	4	6
2 1 1	6	24
2 2 0	8	12
3 1 0	10	24
2 2 2	12	8

The simplest intersection we have found is formed by taking the 32 vectors of norms 6 and 12 and partitioning them into four sets of size 8. The resulting lattices have generator matrices

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & -1 \\ -2 & 2 & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 & 1 \\ -1 & 1 & 2 \\ 2 & -2 & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 2 \\ 2 & -1 & 1 \\ 2 & 2 & -2 \end{pmatrix}, \quad \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ 2 & 2 & 2 \end{pmatrix} \tag{3.16}$$

and their intersection is the fcc lattice[11] with generator matrix $[3,3,0]$, $[3,-3,0]$, $[0,3,-3]$.

Method 4: Using intersections of codes. A fourth method, which however has not yet led to any interesting examples, is to reduce the problem to the analogous question for codes. Let $\Lambda(C)$ denote the lattice obtained by applying

[11] Strictly speaking, we obtain $3D_3$

Construction A to a binary linear code C ([34, Chap. 5]). If $C_1, \ldots, C_r$ are codes of length N whose intersection is a code C, then

$$\Lambda(C) = \Lambda(C_1) \cap \ldots \cap \Lambda(C_r) \ .$$

This can be generalized to nonbinary codes [34, Chaps. 7, 8], in particular to the case where the codes C_i are nonbinary codes whose intersection is binary.

3.4 Analysis of tesselations and Rate-Distortion performance of Periodic Quantizers

As explained in Section 3.1, given a lattice decomposition of Λ as the intersection of lattices $\Lambda^1, \ldots, \Lambda^r$ and given the choice we have taken in this chapter for each individual quantizers Q_{Λ^j}, a periodic tesselation is induced. In order to make it clear, we repeat that the choice that was made in Chapter 2 for each individual quantizer is different and results in different periodic tesselations than the ones obtained in this chapter while keeping the *same* lattice decomposition. The choice that is made in this chapter is expected to give rise to periodic quantizers with better rate-distortion performance while in Chapter 2, the focus is on A/D conversion related applications where $\mathbf{0}$ is never a reconstruction point.

In this Section, we first obtain a general expression for the normalized (dimensionless) second moment associated with these periodic tesselations assuming high-rate quantization and then, we study the tesselations for 4 particular examples, namely, those where the intersection lattices are A_2 (see Fig. 3.1 and 3.2), bcc, fcc, and D_4.

Let the N-dimensional lattice Λ be the intersection of n-dimensional lattices $\Lambda^1, \ldots, \Lambda^r$. The intersections of the Voronoi cells for the Λ^i partition $\mathbb{R}^n$ into a *periodic tesselation.* Let $\mathcal{P}_1, \ldots, \mathcal{P}_k$ be the k representatives for the different polytopes or *cells* that appear in the periodic tesselation. It is important to note that there exist also periodic tesselations which are not based on lattice intersections as the ones considered here which are purely geometrical. The periodicity property has the advantage of allowing a formal and exact high-rate analysis of the rate-distortion performance of these quantizers. The following theorem gives the rate-distortion performance of any periodic quantizer, not necessarily based on lattice intersections, in particular, it can be used to determine the rate-distortion performance of all the periodic quantizers described in this chapter and in Chapter 2. 2.

Theorem 4 *At high rates, the distortion-rate function of an N-dimensional periodic quantizer applied to a source X is given by:*

$$D(R) = G 2^{2h(X)} 2^{-2R}, \quad G = \frac{\sum_{i=1}^{k} \frac{p_i U_i}{V_i}}{N \left(\prod_{i=1}^{k} V_i^{p_i} \right)^{\frac{2}{N}}}\ , \tag{3.17}$$

where the minimal periodic unit of the tesselation has k distinct polytopes $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$, V_i is the volume of the i-th cell, U_i is the unnormalized mean squared error of the i-th cell, $h(X)$ is the differential entropy of X per dimension and $p_i = prob(X \in \mathcal{P}_i)$.

Proof: See Appendix B.2

Note that when there is only one kind of cell (3.17) reduces to the familiar formula

$$G = \frac{U}{NV^{1+\frac{2}{N}}} \tag{3.18}$$

for a lattice quantizer ([31], [34], [55]).

Incidentally, a different expression from (3.17) for the figure of merit was used in a recent paper of Kashyap and Neuhoff [72]. Defining the rate of the quantizer in a different way, results in a different expression in the denominator. However, we believe our formula gives a fairer comparison because the assumptions made in [72] are more restrictive than the ones we make in Theorem 4.

We next describe how the cells of the tesselations were found. This was not included in Chapter 2 because most of the designs that were provided there were in $\mathbb{R}^2$, hence, the cells of the tesselation could be found very easily. In this Chapter, we need to use different geometry rules and methods to make sure that the correct tesselation is found.

Many cells could be found by elementary geometrical reasoning. But in complicated cases we carried out some or all of the following steps:

1. To find the cell containing a point $x \in \mathbb{R}^3$ or $\mathbb{R}^4$ we first quantized $\boldsymbol{x}$ using each of the lattices Λ^j in turn. For each Λ^j, we determined the Voronoi cell C_i containing $\boldsymbol{x}$, or, more precisely, the equations of the hyperplanes bounding C_i.

2. Linear programming (in MATLAB) was then used to determine the vertices of the cell containing $\boldsymbol{x}$. We let $\boldsymbol{w}$ range over a set of 130 points on a sphere centered at $\boldsymbol{x}$ (taken from the tables of spherical codes in [64]), and, for each $\boldsymbol{w}$, we maximized the inner product $\langle \boldsymbol{w}, \boldsymbol{z} \rangle$ subject to the constraints that $\boldsymbol{z}$ lie in the polytope formed by the intersection of *all* the hyperplanes found in (i). Any such solution $\boldsymbol{z}$ is a vertex of the cell, and since the $\boldsymbol{w}$'s are essentially random, the 130 solutions should include all the vertices of the cell.

3. The convex hull program QHULL [5], [1] was used to find the convex hull of these vertices. At this point we have candidates for all the cells in the honeycomb. Since it is theoretically possible (although unlikely) that step (i) might have failed to find all the vertices of a cell, we also verify by hand that the cells fit together to form a proper tiling of the space by checking if the sum of the volumes of the polytopes in the minimal unit is equal to the volume of the minimal unit given by the Voronoi cell of the intersection lattice Λ centered at $\mathbf{0}$. Let $V = det(M_\Lambda)$ be the volume of a fundamental region or Voronoi cell for Λ and n_i be the number of cells of type $\mathcal{P}_i$ with volume V_i that are contained in each Voronoi cell of Λ. Then, we check the volume equation:

$$V = n_1 V_1 + \ldots + n_k V_k \ . \tag{3.19}$$

4. The XGobi program [104] for displaying multi-dimensional data was used to help visualize the cells and their neighbors.

To compute the volumes and second moments of the cells we decomposed the cells into simplices and used the formulae in [31] and [34, Chap. 21].

We now begin our study of the periodic tesselations formed by some of the intersections described in Section 3.3. In order to represent the tesselations in a compact way, we use an undirected graph where each node is represented by a circle, as shown in Fig. 3.3. A circle containing i refers to a cell of type $\mathcal{P}_i$, and

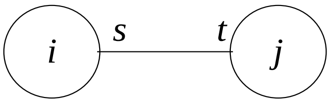


Figure 3.3: Undirected graph representing the incidence between cells $\mathcal{P}_i$ and $\mathcal{P}_j$

an edge between i and j indicates that $\mathcal{P}_i$ and $\mathcal{P}_j$ share a common face of the maximal dimension (here 1), and the edge labels indicate that each $\mathcal{P}_i$ is adjacent to s cells of type $\mathcal{P}_j$ and each $\mathcal{P}_j$ to t cells of type $\mathcal{P}_i$.

3.4.1 The hexagonal lattice as an intersection of three lattices

The hexagonal lattice is the intersection of the three rectangular lattices given by (3.2), as in Fig. 3.1. The periodic tesselation is shown in Fig. 3.2. We now compute the mean squared error for this quantizer.

There are four types of cells. The origin is contained in a hexagon $\mathcal{P}_1$ (horizontally shaded in Fig. 3.2) of edge length $1/\sqrt{3}$, area $V_1 = \sqrt{3}/2$ and second moment $U_1 = 5\sqrt{3}/72$. The second type of cell is a small equilateral triangle $\mathcal{P}_2$ (cross-hatched), with $V_2 = \sqrt{3}/12$, $U_2 = \sqrt{3}/432$. $\mathcal{P}_3$ is an isosceles triangle (diagonally shaded), with $V_3 = \sqrt{3}/12$, $U_3 = 5\sqrt{3}/1296$. The fourth type, $\mathcal{P}_4$, is a larger equilateral triangle (vertically shaded), with $V_4 = \sqrt{3}/4$, $U_4 = \sqrt{3}/48$.

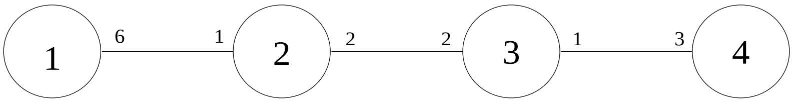


Figure 3.4: Incidences between cells in periodic tesselation for hexagonal lattice.

The incidences between the different types of cells are shown in Fig. 3.4. The Voronoi cell $\mathcal{V}$ for the intersection lattice is enclosed by the broken lines in Fig. 3.2. In the notation of Section 3.4, $\mathcal{V}$ contains $n_1 = 1$ copy of $\mathcal{P}_1$, $n_2 = 6$ copies of $\mathcal{P}_2$, $n_3 = 6$ copies of $\mathcal{P}_3$ and $n_4 = 6 \times \frac{1}{3} = 2$ copies of $\mathcal{P}_4$, and the volume equation (3.19) reads

$$2\sqrt{3} = \frac{\sqrt{3}}{2} + \frac{\sqrt{3}}{2} + \frac{\sqrt{3}}{2} + \frac{\sqrt{3}}{2}\ . \tag{3.20}$$

Thus the probabilities $p_1, \ldots, p_4$ of a randomly chosen point in the plane belonging to a cell of each type are all equal to 1/4. From (3.17), the normalized mean squared error for this quantizer is $G = 2^{7/4}/27 = 0.1246\ldots$ This value is considerably worse than the value $0.080188\ldots$ for the hexagonal lattice itself.

3.4.2 The bcc lattice as an intersection of three lattices

The bcc lattice is the intersection of the three "rectangular" lattices $\Lambda^1, \Lambda^2, \Lambda^3$ defined in (3.14). Each Λ^j is congruent to $\mathbb{Z} \times \sqrt{2}\mathbb{Z} \times \sqrt{2}\mathbb{Z}$, and has as Voronoi cell a brick with square cross-section. There is an obvious symmetric group S_3 that permutes the Λ^j.

Again the periodic tesselation contains four types of cells. The origin is contained in the intersection of the Voronoi cells at 0 for the three Λ^i. This is a cube, $\mathcal{P}_1$, with vertices $(\pm 1/2, \pm 1/2, \pm 1/2)$, volume $V_1 = 1$, and second moment

$U_1 = 1/4$. Across each square face of $\mathcal{P}_1$ is a square pyramid $\mathcal{P}_2$, such as that with base $[1/2, \pm 1/2, \pm 1/2]$, apex $[1, 0, 0]$, $V_2 = 1/6$, $U_2 = 11/600$. Across each triangular face of $\mathcal{P}_2$ is a tetrahedron $\mathcal{P}_3$, such as that with vertices $[1/2, \pm 1/2, 1/2]$, $[1, 0, 0]$, $[1, 0, 1/2]$, $V_3 = 1/24$, $U_3 = 1/512$. Finally, across the other three faces of $\mathcal{P}_3$ we reach a fourth type of cell, $\mathcal{P}_4$, a quarter-octahedron, which occurs in two orientations, one having vertices such as

$$\left[\frac{1}{2}, 0, \frac{1}{2}\right], \quad \left[\frac{1}{2}, 0, 1\right], \quad \left[1, 0, \frac{1}{2}\right], \quad [1, 0, 1], \quad \left[\frac{1}{2}, \pm\frac{1}{2}, \frac{1}{2}\right], \tag{3.21}$$

the other having vertices such as

$$\left[1, \frac{1}{2}, \frac{1}{2}\right], \quad \left[1, 0, \frac{1}{2}\right], \quad \left[1, \frac{1}{2}, 0\right], \quad [1, 0, 0], \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right], \left[\frac{3}{2}, \frac{1}{2}\frac{1}{2}\right]. \tag{3.22}$$

These may be described as quarters of squat octahedra. E.g., (3.21) is a quarter of the octahedron with vertices $[\pm 1, 0, \pm 1]$, $[1/2, \pm 1/2, 1/2]$. For $\mathcal{P}_4$ we have $V_4 = 1/12$, $U_4 = 1/192$.

No further types of cell appear: every face of either version of $\mathcal{P}_4$ leads to a $\mathcal{P}_3$. The incidence diagram is shown in Fig. 3.5.

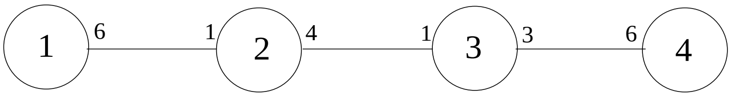


Figure 3.5: Incidences between cells for periodic tesselation for bcc lattice.

The Voronoi cell $\mathcal{V}$ for the intersection lattice is a truncated octahedron with 24 vertices $[0, \pm 1/2, \pm 1]$. This contains $n_1 = 1$ copy of $\mathcal{P}_1$, $n_2 = 6$ copies of $\mathcal{P}_2$ and $n_3 = 24$ copies of $\mathcal{P}_3$. The cells of type $\mathcal{P}_4$ partially overlap $\mathcal{V}$. There are 12 of type (3.21), intersecting $\mathcal{V}$ in a tetrahedron such as that with vertices $[1/2, \pm 1/2, 1/2]$, $[1/2, 0, 1]$, $[1, 0, 1/2]$, with volume 1/24. There are also 24 of type (3.22), intersecting $\mathcal{V}$ in a tetrahedron such as $[1/2, 1/2, 1/2]$, $[1, 0, 0]$, $[1, 0, 1/2]$, $[1, 1/2, 0]$, with volume 1/48. The volume equation (3.19) reads

$$\begin{aligned} 4 &= 1 \times 1 + 6 \times \frac{1}{6} + 24 \times \frac{1}{24} + 12 \times \frac{1}{24} + 24 \times \frac{1}{48} \\ &= 1 + 1 + 1 + 1 \ , \end{aligned} \tag{3.23}$$

so again the probabilities p_i of a randomly chosen point belonging to a cell of given type are all equal to 1/4. The normalized mean squared error is

$$G = \frac{751\sqrt{3}}{9600} = 0.1355\ldots$$

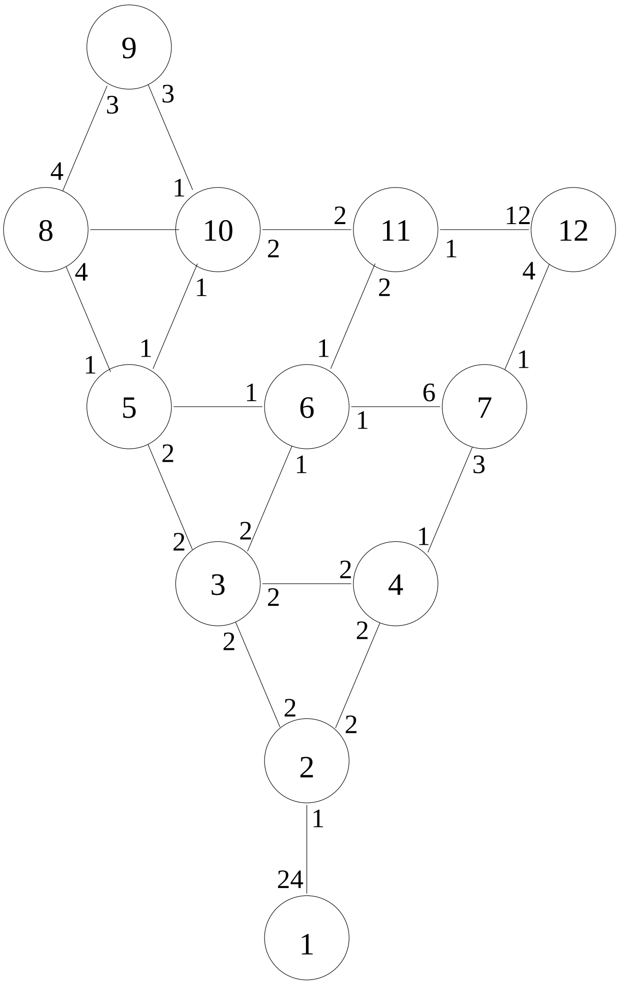


Figure 3.6: Incidences among cells of fcc periodic tesselation.

3.4.3 The fcc lattice as an intersection of four lattices

The fcc lattice is the intersection of the four lattices $\Lambda^1, \ldots, \Lambda^4$ defined in (3.16).

Each of these has Gram matrix equivalent to

$$\begin{pmatrix} 6 & -3 & 0 \\ -3 & 6 & 0 \\ 0 & 0 & 12 \end{pmatrix}$$

and is a direct sum $\sqrt{3}A_2 \oplus \sqrt{12}\mathbb{Z}$, with Voronoi cell a hexagonal prism. The Λ^j look more symmetrical if they are written in the coordinates used to describe the root lattice A_3 ($\cong D_3$), that is, using four coordinates that add to 0. Then Λ^1 has generator matrix

$$\begin{pmatrix} 0 & 2 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 3 & -1 & -1 & -1 \end{pmatrix}$$

and the others are given by cyclic shifts of these columns. This shows that there is a symmetric group S_4 permuting the Λ^j. However, the three-dimensional coordinates given in (3.16) are more convenient for computations.

This periodic tesselation is the most complicated we have analyzed and we shall give only a brief description. There are twelve types of cells, $\mathcal{P}_1, \ldots, \mathcal{P}_{12}$, whose parameters are summarized in Table 3.3 and whose incidences are shown in Fig. 3.6. Fig. 3.7 and 3.8 shows cross-sections through the periodic tesselation along the planes $z = 0$ and $z = 0.35$.

The following is a brief description of the cells, including coordinates for one cell of each type.

$\mathcal{P}_1$. Obtained from cube by pushing in corners and pulling out edge midpoints. Vertices: all cyclic shifts and sign changes of $[3/2, 0, 0]$, $[1, 1, 0]$, $[3/4, 3/4, 3/4]$.

$\mathcal{P}_2$. Pyramid with kite-shaped base. Base: $[3/2, 0, 0]$, $[1, 1, 0]$, $[1, 0, 1]$, $[3/4, 3/4, 3/4]$, apex: $[3/2, 1/2, 1/2]$.

i	v	e	f	n_i	V_i	p_i	U_i
1	26	48	24	1	9	1/6	1449/160
2	5	8	5	24	1/8	1/18	41/3200
3	6	12	8	24	1/4	1/9	9/320
4	5	8	5	24	1/8	1/18	41/3200
5	5	9	6	24	1/10	2/45	427/50000
6	4	6	4	48	1/40	1/45	443/320000
7	10	18	10	8	27/40	1/10	41013/160000
8	6	12	8	6	1	1/9	47/180
9	5	9	6	8	3/8	1/18	189/3200
10	4	6	4	24	1/40	1/90	1897/1280000
11	5	8	5	24	3/40	1/30	2319/400000
12	22	36	16	2	63/10	7/30	213597/40000

Table 3.3: The twelve types of cells in the fcc case, showing numbers of vertices, edges, faces (v, e, f), the number per fcc cell (n_i), and their volumes, probabilities and second moments(V_i, p_i, U_i).

$\mathcal{P}_3$. Irregular octahedron with vertices $[3/2, 0, 0]$, $[3/2, 3/2, 0]$, $[1, 1, 0]$, $[2, 1, 0]$, $[3/2, 1/2, \pm 1/2]$.

$\mathcal{P}_4$. Congruent to $\mathcal{P}_2$. Vertices: $[3/2, 0, 0]$, $[3/4, 3/4, 3/4]$, $[3/2, 1/2, 1/2]$, $[1/2, 3/2, 1/2]$, $[1, 1, 0]$.

$\mathcal{P}_5$. Irregular polyhedron with six faces. Vertices: $[3/2, 0, 0]$, $[2, 1, 0]$, $[2, 0, 1]$, $[9/5, 3/5, 3/5]$, $[3/2, 1/2, 1/2]$.

$\mathcal{P}_6$. Tetrahedron: $[2, 1, 0]$, $[3/2, 3/2, 0]$, $[3/2, 1/2, 1/2]$, $[9/5, 3/5, 3/5]$.

$\mathcal{P}_7$. "Flying saucer": hexagonal base (cyclic shifts of $[3/2, 3/2, 0]$ and $[9/5, 3/5, 3/5]$) with four vertices above it (cyclic shifts of $[3/2, 1/2, 1/2]$ and $[3/4, 3/4, 3/4]$ at apex).

$\mathcal{P}_8$. Irregular octahedron with vertices $[2, \pm 1, 0]$, $[2, 0, \pm 1]$, $[3/2, 0, 0]$, $[3, 0, 0]$.

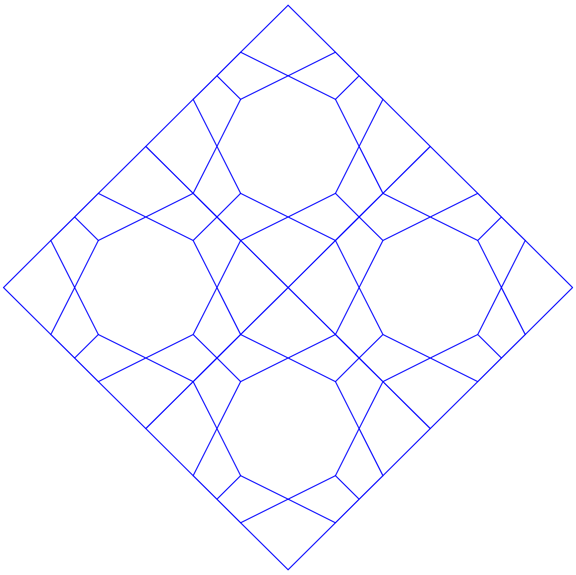


Figure 3.7: Cross-section of the periodic tesselationf for the fcc case along plane $z = 0$ ($-3 \leq x \leq 9$, $-6 \leq y \leq 6$), with origin at center of octagon on left. Only three cells are visible: $\mathcal{P}_1$ (octagon), $\mathcal{P}_2$ (small kite), $\mathcal{P}_9$ (large dart).

$\mathcal{P}_9$. Regular tetrahedron (vertices $[2, 1, 0]$, $[2, 0, 1]$, $[3, 0, 0]$, $[3, 1, 1]$) with triangular cap (apex $[9/4, 3/4, 3/4]$) on one face.

$\mathcal{P}_{10}$. Irregular tetrahedron with vertices $[2, 1, 0]$, $[2, 0, 1]$, $[9/4, 3/4, 3/4]$, $[9/5, 3/5, 3/5]$.

$\mathcal{P}_{11}$. Another pyramid with kite-shaped base. Base: $[3/2, 3/2, 0]$, $[9/4, 3/4, 3/4]$, $[9/5, 3/5, 3/5]$, $[12/5, 6/5, 3/5]$, apex: $[2, 1, 0]$.

$\mathcal{P}_{12}$. See Fig. 3.9. Has 26 vertices, four hexagonal and 12 kite-shaped faces. Vertices are all permutations of $[3/2, 3/2, 0]$, $[3/2, 3/2, 3]$, $[9/4, 3/4, 3/4]$, $[9/4, 9/4, 9/4]$, $[9/5, 3/5, 3/5]$, $[12/5, 9/5, 9/5]$, $[12/5, 6/5, 3/5]$.

If we decompose $\mathbb{R}^3$ into Voronoi cells for the intersection lattice Λ (3.20), just three of the twelve types of cells are cut by the boundary walls. Cells of type

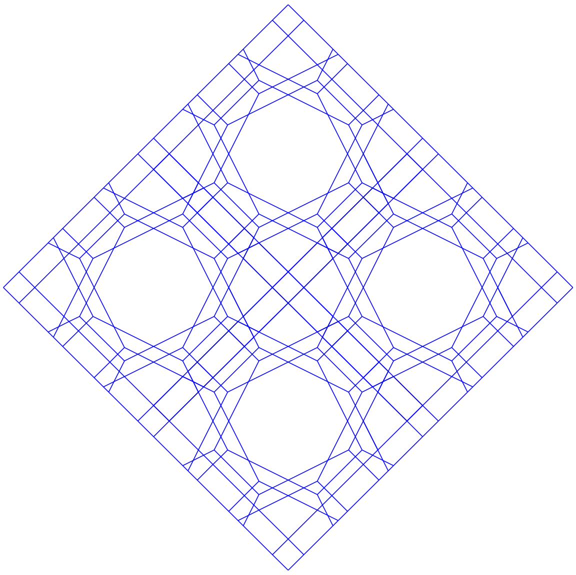


Figure 3.8: Cross-section through the periodic tesselation for the fcc case along plane $z = 0.35$. Cross-sections of all 12 types of cells can be seen.

$\mathcal{P}_9$ are cut into three equal pieces, cells of type $\mathcal{P}_{11}$ are cut in half, and cells of type $\mathcal{P}_{12}$ are cut into four equal ice-cream cone shaped pieces. The base of each cone is at the center of $\mathcal{P}_{12}$, and the top contains one of the hexagons and parts of the neighboring faces.

The normalized mean squared error is

$$G = \frac{12269777}{816480000} \, 3^{28/135} \, 5^{8/27} \, 7^{38/45} = 0.1572\ldots .$$

3.4.4 The D_4 lattice as an intersection of three cubic lattices

The three cubic lattices Λ^1, Λ^2, Λ^3 are defined in (3.5) and their intersection $\Lambda \cong D_4$ in (3.6). There are just four types of cells, $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, $\mathcal{P}_4$, whose

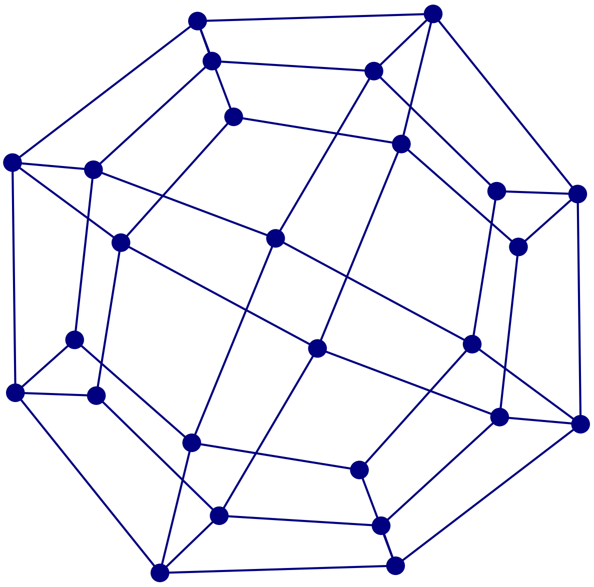


Figure 3.9: 26-vertex cell $\mathcal{P}_{12}$.

properties are summarized in Table 3.4 and whose intersections are shown in Fig. 3.10. We use coordinates $[a, b, c, d]$ for points in $\mathbb{R}^4$.

i	v	f	n_i	V_i	p_i	U_i
1	24	24	1	8	1/4	104/15
2	7	9	24	1/3	1/4	8/105
3	5	5	96	1/12	1/4	11/900
4	6	9	32	1/4	1/4	1/20

Table 3.4: Cells in periodic tesselation for the D_4 case, showing numbers of vertices and 3-dimensional faces (v, f), the number per D_4 cell (n_i) and their volumes, probabilities and second moments (V_i, p_i, U_i).

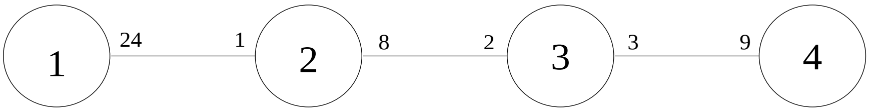


Figure 3.10: Incidences among cells of D_4 honeycomb.

$\mathcal{P}_1$ is the 4-dimensional regular polytope known as a 24-cell ([34], [37]). The Voronoi cells for Λ^1, Λ^2, Λ^3 at the origin are all cubes, whose intersection is bounded by the hyperplanes

$$|a| \le 1,\ |b| \le 1,\ |c| \le 1,\ |d| \le 1,\ |a| + |b| + |c| + |d| \le 1,$$

which is the 24-cell with vertices of the form $[\pm 1, \pm 1, 0, 0]$.

Across each of the 24 octahedral faces of $\mathcal{P}_1$ we reach an octahedral-based pyramid $\mathcal{P}_2$, such as that with vertices $[1, \pm 1, 0, 0]$, $[1, 0, \pm 1, 0]$, $[1, 0, 0, \pm 1]$ and $[2, 0, 0, 0]$ (the apex).

There are eight other faces of $\mathcal{P}_2$, regular tetrahedra; these lead to copies of $\mathcal{P}_3$, which is an irregular simplex such as that with vertices $[1, 1, 0, 0]$, $[1, 0, 1, 0]$, $[1, 0, 0, 1]$, $[2, 0, 0, 0]$, $[1, 1, 1, 1]$.

Finally, three of the five faces of each $\mathcal{P}_3$ lead to cells of the fourth type, $\mathcal{P}_4$. This can best be described as the product of two skew equilateral triangles of different sizes (just as a tetrahedron in three dimensions is the product of two skew line segments). Take an equilateral triangle with vertices $p = [2, 1, 1, 0]$, $q = [1, 1, 0, 0]$, $r = [1, 0, 1, 0]$ and another with vertices $P = [2, 0, 0, 0]$, $Q = [1, 1, 1, 1]$,

$R = [1, 1, 1, -1]$. Then $\mathcal{P}_4$ is their convex hull. There are nine tetrahedral faces given by the convex hull of an edge of the first triangle and an edge of the second triangle.

If we decompose $\mathbb{R}^4$ into Voronoi cells for the intersection lattice, only cells of type $\mathcal{P}_4$ are cut by the boundary walls. Each $\mathcal{P}_4$ is divided into three equal pieces, a typical piece being an "ice-cream cone" whose center is at the center of $\mathcal{P}_4$ and whose three-dimensional face is the convex hull of the second triangle (P, Q, R) and any edge of the first triangle.

The volume equation (3.19) then reads

$$32 = 1 \times 8 + 24 \times \frac{1}{3} + 96 \times \frac{1}{12} + 96 \times \frac{1}{4} \times \frac{1}{3} \,.$$

Again a random point is equally likely to fall into a cell of any of the four types. The normalized mean squared error is

$$G \;=\; \frac{757}{8400}\, 2^{1/8}\, 3^{1/4} \;=\; 0.1293\ldots \;.$$

3.5 Conclusions and comments

The basic results presented in this chapter are as follows. We fist study general methods in order to write a good lattice Λ as the intersection of a set of simple lattices $\Lambda^1, \ldots, \Lambda^r$. Then, we obtain concrete decompositions for the cases where $\Lambda = A_2, A_3^*$ (bcc), A_3 (fcc), $D_4, E_6^*, E_8, K_{12}, \Lambda_{24}$ (Leech), BW_N (Barnes-Wall). Then, we derive a formula that characterizes the rate-distortion performance of any periodic periodic quantizers in terms of a generalized normalized second moment G. Next, we study in full detail the tesselations that are obtained for the given constructions in the cases where $\Lambda = A_2, A_3^*$ (bcc), A_3 (fcc), D_4, showing all the cells of the tesselations and their incidences and calculating the corresponding values of G.

In each of the honeycombs of Sections 3.4.1, 3.4.2 and 3.4.4 just four types of cells occurred. This is easily explained in the case of the D_4 honeycomb: there are three equivalent lattices Λ^1, Λ^2, Λ^3, and the associated quantizers are essentially making binary decisions about the location of a point with respect to the intersection lattice. So the space is divided up into regions that can be labeled 000, 001, 011 and 111.

This argument does not quite apply to the A_2 or bcc honeycombs, since there the individual lattices themselves are not fully symmetric (rectangular rather than square in the A_2 case, for instance). So it is fortuitous that only four cells occur. In contrast, the fcc honeycomb shows that the number of cells can increase rapidly in less fortunate cases with more component lattices.

From a rate-distortion point of view, as for practical quantization related applications, further research needs to be done in order to find competitive quantizers. On the one hand, it is possible that better quantizers could be obtained by amalgamating (merging) less symmetrical cells. For example, in Fig. 3.2, the diagonally and vertically shaded triangles could be amalgamated to give a honeycomb made up of regular hexagons and equilateral triangles with the same edge length as the hexagons. The new honeycomb will have larger absolute error but a smaller normalized error G. We did not investigate this possibility for the different lattice decompositions we have obtained. On the other hand, another interesting topic for future research is the design of quantizers by taking a simple initial lattice Λ^1 and combining it with several other lattices which are both rotations *and translations* of Λ^1.

From a complexity point of view, it is interesting to mention that in some cases, specially when the number of simple lattices that constitute the decomposition of a complicated lattice is small, it is faster to compute $(Q_{\Lambda^1}, Q_{\Lambda^2}, \ldots, Q_{\Lambda^r})$ than Q_Λ. On the other hand, the constructions presented here are still useful for quantizing overcomplete expansions in $\mathbb{R}^N$.

Chapter 4

Oversampled Steerable Transforms: Quantization and Rotation Invariance[*]

4.1 Introduction and Motivation

Feature detection and extraction is a very important step in several applications (e.g., classification, content based retrieval, image understanding systems, etc...) where features of interest should be preserved in the representation that is used. For instance, consider the application of content-based access to databases containing large amounts of multimedia data, where text-based indexing is not sufficient. The images in a database are normally compressed using either a DCT or a (orthogonal or biorthogonal) wavelet based algorithm, because of their critical

[*] The publications related to this chapter are [9, 88, 13, 14].

sampling as opposed to overcomplete decompositions. Orthogonal and biorthogonal critically sampled transforms, like all of the linear transforms commonly used in image compression, have important drawbacks in the representation they give rise to: a) lack of shift and rotation invariance because the representation is highly dependent on the relative alignment of the image and the subsampling lattices b) the selectivity in orientation is limited, e.g., all the $2D$ filters which are built from the outer product of $1D$ filters, can only only detect energy (information) in three orientations: horizontal (0 degrees), vertical (90 degrees) and diagonals (45 and 135 degrees), as shown in Fig. 4.1. Although it is possible to increase the directionality while keeping critical sampling by using some alternative filter designs [3, 4] which are not based on outer products, the *critical sampling* in these transforms makes it impossible to achieve shift or rotation invariance. The main goal in our work is to achieve rotation invariance in the context of a content based image retrieval application. Any typical content based retrieval system consists of two main tasks, a texture extraction process and a similarity measurement scheme. Most texture extraction methods for retrieval consist of first

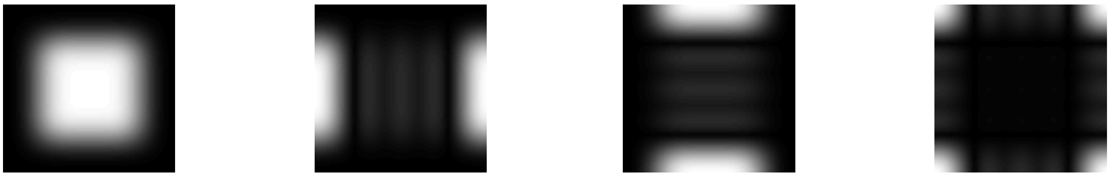


Figure 4.1: Set of 2D filters obtained from the 1D Daubechies orthogonal filter bank 'daub3'

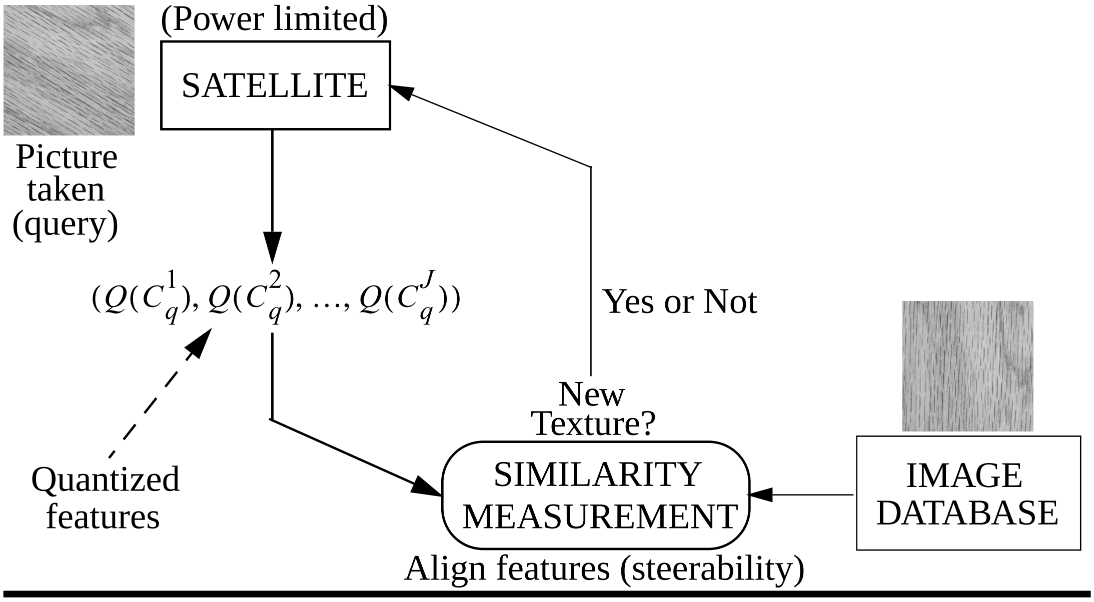


Figure 4.2: Rotation-Invariance in remote content based image retrieval

filtering the signal with some filter banks and then measuring the energies (possibly weighted) of the corresponding output subbands as the extracted features. Consider the problem illustrated in Fig. 4.2 where a satellite is capturing images which may be rotated versions of images already present in the image database on the earth. Bandwidth and power are very limited for the satellite and this means that it should be avoided as much as possible sending images to the earth which are unnecessary, that is, which belong to a class for which the database has already samples of it. In the context of content based image retrieval, this would be an example of a query search by example. Instead of sending images in a continuous manner, a set of representative features describing the new image are first obtained and transmitted (after being quantized) to the earth. On the earth, some similarity measurement is performed between these features and the features of the images contained in our database. Suppose that the new image is a rotated version of an image already present in the database. In this case, it is required to perform an alignment between the features of the 2 textures being compared. In order to perform this alignment, we need to define a set of features which can be somehow rotated as we rotate the underlying image. We call this kind of features "steerable" features. As it is shown in this chapter, these desirable steerable features can be only defined using an oversampled filter bank system called steerable filter bank.

As it is illustrated in the example given by Fig. 4.2, a set of useful features for discrimination purposes in images can be obtained by detecting information of an image in different orientations. A brute-force and very inefficient approach to do this would be to use a large set of filters, each being oriented in a different direction. Designing all these filters is not necessary if filters steerable under rotation are available. A filter is called steerable under some transformation Lie group (e.g. translation, rotation, scaling, etc...) if transformed versions of this filter can always be expressed as a linear combination of a fixed, finite set of basis filters. For the particular case of steerability under rotation, a set of basis filters can be applied to an image and since convolution is linear, we can calculate exactly, by linearly combining these basic responses, the filtering of that image at an arbitrary orientation, without explicitly designing and applying different filters for each of the desired orientations. This has been proved to be very useful in many different vision and image processing tasks, such as segmentation or texture analysis [52, 101]. This powerful representation turns out to be necessarily redundant or overcomplete.

The similarity measurement usually consists of some norm-based distance calculated in the feature space. Since we want to achieve rotation-invariance, we need to extract features which are *steerable* in the sense that given two rotated versions of the same image, the features from one version can be mapped to the other version in a simple way. As we show in this chapter, in order to extract features having this property, it is necessary to use a steerable transform which is an overcomplete representation and a regular wavelet transform is not adequate.

On the other hand, notice also that in the context of feature extraction over many different compressed images, if we were using a critically sampled transform, we first would have to decompress each image and then, we would have to apply a steerable transform to the decompressed image. Using directly a steerable transform to code the images, we can extract all these many different features directly in the transformed domain. Therefore, in cases where many different features have to be extracted and very multi-purpose and detailed image queries are needed, critically sampled transforms may not be the best option because they would involve a high time complexity due to the large number of decompression operations that would have to be performed.

Given this requirement, one of the problems that we study in this chapter is how to achieve as much compression efficieny as possible using these oversampled steerable representations. Second, we study in detail how to solve the problem of rotation invariance using energy-related features measured from the steerable transform subbands. We focus our study on transforms which are steerable under rotation although some of the theory we introduce is also valid for any transformation Lie group. We analyze angular oversampling (e.g. increasing the number of orientations) in the context of steerable transforms which have not considered by prior research and explore techniques to represent efficiently this oversampled data. The angular oversampling or oversteering is also motivated because it allows us to establish some "consistency" constraints [39, 60, 84, 111] on the coefficients of a steerable representation with many orientations (oversteered representation), which reduces the amount of information lost in the quantization process and thus, an increase in oversampling, results in an increase of the accuracy and resolution of the corresponding transform coefficients.

In Section 4.2 we first provide a complete review of the basic concepts, properties and construction of functions which are steerable under a Lie Transformation group, focusing on the fundamental concepts of basis functions, steering functions, equivariant spaces, interpolation equation, infinitesimal generator and tangent space, which, as illustrated in Section 4.2.1 and Section 4.2.2 provides a general method for constructing $1D$ equivariant function spaces for the 1-parameter translation group, using simple linear algebra, which is the basis to construct a $2D$ digital filter bank which is steerable under rotation.

Next, we explain our different approaches in order to represent the oversampled data, placing an emphasis on how to decrease the error on the transform coefficients after quantization has taken place when the number of orientations that are used is increased. In Section 4.4.1, we show that increasing the number of orientations results in a better energy compaction in angle. In Section 4.4.2, we describe two methods which reduce the quantization error in the transform coefficients by establishing two "consistency" constraints, one due to the smoothness (steerability) constraints on the steerable curve linking all the angular coefficients of all the subbands and another one due to the quantization itself. We explain two ways of using these two consistency constraints, one based on projection on convex sets (POCS) and another one based on calculating regions of uncertainty, and provide some experimental results supporting our approach.

Then, in Section 4.5 we concentrate on the problem of rotation invariance in the context of content-based image retrieval. We propose a steerable transform that serves as a basis for an image retrieval system which can recognize the situation where the query image is a rotated version of some image already present in the database. In order to achieve this goal, we define some *steerable features* which make use of correlations between different orientations within each level in addition to the energy in each orientation. Our similarity measurement basically aims at aligning the features between two different textures by rotating the features so that if two texture samples are rotated versions of the same texture class, the distance, after the alignment, is as small as possible and thus, the equivalence between two samples of the same texture can be recognized. Several experiments are shown illustrating our proposed similarity measurement using steerable features. We also compare our results with those obtained using a wavelet pyramid.

4.2 Review of basic Definitions, Properties and Construction of Steerable transforms

The original definition of steerability was proposed by Freeman and Adelson [52] for the particular case of rotation. Simoncelli et al. [100] extended this definition to include translation and scaling and later Perona [89, 90] used the term "deformable" to refer to functions which are steerable under arbitrary compact transformations. Since most of the typical transformations encountered in signal processing (e.g., translation, rotation, scaling) are Lie transformation groups, we consider only Lie groups.

In this section, we provide a complete overview of the general formulation for the construction of steerable functions based on Lie Theory. The main reason for which we use a Lie Theory formulation is that the derivation of steerable functions is much clearer, simpler and much more elegant than all the previous formulations provided in [52, 100, 89, 90].

Definition 15 *A family of transformations $\{g(\tau_1, \tau_2, \ldots, \tau_k)\}$ parameterized by $\tau_1, \ldots, \tau_k$ over some predefined range and acting on the coordinates of a (not necessarily real) function $f(\boldsymbol{x}) : \mathbb{R}^N \mapsto \mathbb{C}$ is a Lie group G if the following properties are satisfied: (1) it satisfies the algebraic group conditions of closure under composition, associativity and the existence of an inverse and identity, and (2) the maps for inverse and composition are smooth (infinitely differentiable).*

For instance, for a $2D$ function $f(x_1, x_2)$, a well known family of transformations is given by the 1-parameter group of rotations $g_R(\tau)$ in the plane such that $g_R(\tau)\ f(x_1, x_2) = f(x_1 \cos(\tau) - x_2 \sin(\tau), x_1 \sin(\tau) + x_2 \cos(\tau))$. Notice that a transformation group implies a coordinate transformation $x_i' = s_\tau(x_i)$ for each of the coordinates. In the previous example, for instance, $x_1' = x_1 \cos(\tau) - x_2 \sin(\tau)$. Another familiar example is the 2-parameter translations, that is, $g_{t_{x_1}, t_{x_2}}(\tau_1, \tau_2)$ $f(x_1, x_2) = f(x_1 - \tau_1, x_2 - \tau_2)$. The mathematical treatment of Lie theory in this chapter is rather limited, that is, we only present the necessary concepts. For a much more detailed exposition of Lie theory, see the numerous books about the subject [28, 15, 23, 69, 95, 73, 75, 106, 86].

Next, we give a general definition of steerable function which includes any Lie transformation group. This definition is adapted from [67, 68].

Definition 16 *A function $f(\boldsymbol{x}) : \mathbb{R}^N \mapsto \mathbb{C}$ is steerable under a k-parameter Lie transformation group G if any transformation $g(\boldsymbol{\tau}) \in G$ of $f(\boldsymbol{x})$ can be written as a linear combination of a fixed, finite sef of basis functions $\{\psi_i(\boldsymbol{x})\}_{i=1}^J$:*

$$g(\boldsymbol{\tau})\ f(\boldsymbol{x}) = \sum_{i=1}^{J} \alpha_i(\boldsymbol{\tau})\psi_i(\boldsymbol{x}) = \boldsymbol{\alpha}^T(\boldsymbol{\tau})\boldsymbol{\Psi}(\boldsymbol{x}) \tag{4.1}$$

where $\boldsymbol{\Psi}(\boldsymbol{x}) = [\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x}), \ldots, \psi_J(\boldsymbol{x})]^T$, the vector $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_k]^T$ parameterizes the family of transformations in the group G, the vector $\boldsymbol{\alpha} = [\alpha_1(\boldsymbol{\tau}), \alpha_2(\boldsymbol{\tau}), \ldots, \alpha_J(\boldsymbol{\tau})]$ contains the set of steering functions and the vector $\boldsymbol{\Psi}(\boldsymbol{x})$ contains the basis functions.

The steering functions $\{\alpha_i(\boldsymbol{\tau})\}_{i=1}^J$ depend only on the k transformation parameters $\{\tau_i\}_{i=1}^k$ and are unique given a particular function $f(\boldsymbol{x})$ and the particular set of basis functions $\boldsymbol{\Psi}$ that is needed to steer $f(\boldsymbol{x})$. It is clear that for a given function $f(\boldsymbol{x})$, the set of basis functions is not unique because any (non-singular) linear transformation of the set of basis functions could also be used. Let us assume that J is the minimum number of basis functions required and theferore, that the set of functions $\{\psi_i(\boldsymbol{x})\}_{i=1}^J$ are linearly independent. It is also important to mention that not all functions can be analytically steerable, that is, they may require an infinite number of basis functions[2].

[2] In this case, the goal is, for a given k, to find the best set of k basis functions which give the minimum error in achieving exact steerability [110].

As a simple illustrative example, consider the $2D$ function $f(x_1, x_2)$ (e.g. $\boldsymbol{x} = [x_1, x_2]$) given by the first x_1-derivative $G'_{x_1}(x_1, x_2)$ of a $2D$ gaussian $G(x_1, x_2) = e^{-(x_1^2+x_2^2)}$. $G'_{x_1}(x_1, x_2) = -2x_1 e^{-(x_1^2+x_2^2)}$ is steerable under the one-parameter group of rotations where the set of basis functions is given by $\boldsymbol{\Psi}(x_1, x_2) = [G'_{x_1}(x_1, x_2), G'_{x_2}(x_1,\ x_2)]^T$ and the set of steering functions is simply $\boldsymbol{\alpha}(\tau) = [\cos(\tau), \sin(\tau)]^T$. Thus, we have that $G'_{\tau}(x_1, x_2) = g(\tau)\ G'_{x_1}(x_1, x_2) = \cos(\tau)$ $G'_{x_1}(x_1, x_2) + \sin(\tau) G'_{x_2}(x_1, x_2)$, that is, the directional derivative on the plane along the direction corresponding to an angle τ, can be obtained through a linear combination of the derivatives along 0 (x_1-derivative) and 90 (x_2-derivative) degrees. Notice that if we choose any two angles τ_1 and τ_2 ($\tau_1 \neq \tau_2$), we can express both $\psi_1(x_1, x_2) = G'_{x_1}(x_1, x_2)$ and $\psi_2(x_1, x_2) = G'_{x_2}(x_1, x_2)$ as follows:

$$\begin{pmatrix} \psi_1(x_1, x_2) \\ \\ \psi_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} G'_{x_1}(x_1, x_2) \\ \\ G'_{x_2}(x_1, x_2) \end{pmatrix} = \begin{pmatrix} \cos(\tau_1) & \sin(\tau_1) \\ \\ \cos(\tau_2) & \sin(\tau_2) \end{pmatrix}^{-1} \begin{pmatrix} G'_{\tau_1}(x_1, x_2) \\ \\ G'_{\tau_2}(x_1, x_2) \end{pmatrix} \tag{4.2}$$

Since $G'_{\tau_1}(x_1, x_2)$ and $G'_{\tau_2}(x_1, x_2)$ are themselves steerable with steering functions $G'_{x_1}(x_1, x_2)$ and $G'_{x_2}(x_1, x_2)$, this implies directly that each basis function is itself steerable with the same set of basis functions although with different steering functions. Since this is always true in general and in both directions, it makes more sense to define equivalently the concept of steerability in terms of the function space spanned by the basis functions $\{\psi_i(\boldsymbol{x})\}$.

Definition 17 *An N-dimensional function space $V = span\{\psi_1(\boldsymbol{x}), \ldots, \psi_J(\boldsymbol{x})\}$ is equivariant under a Lie transformation group G if every $\psi_i(\boldsymbol{x})$ is steerable with respect to the basis functions $\{\psi_1(\boldsymbol{x}), \ldots, \psi_J(\boldsymbol{x})\}$, that is, there exists a matrix function $\boldsymbol{A}(\boldsymbol{\tau})$ (interpolation matrix) such that the following interpolation equation holds:*

$$g(\boldsymbol{\tau})\ \boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{\tau})\boldsymbol{\Psi}(\boldsymbol{x}) \quad \forall\ g(\boldsymbol{\tau}) \in G \tag{4.3}$$

For the previous example, the two-dimensional function space $V = span\{G'_x(x,y), G'_y(x,y)\}$ is equivariant and the interpolation matrix $\boldsymbol{A}(\theta)$ is just a pure rotation matrix. The equivariance property means that the function space V is invariant (satisfies closure) under the associated transformation group G and a function $f(\boldsymbol{x})$ is steerable under a k-parameter transformation group if and only if it belongs to some function space which is equivariant under the same transformation group. Notice that any function $f(\boldsymbol{x}) \in V$ with $f(\boldsymbol{x}) = \sum_i b_i\psi_i(\boldsymbol{x}) = \boldsymbol{b}^T\boldsymbol{\Psi}(\boldsymbol{x})$ is steerable with the same set of basis functions $\{\psi_i(\boldsymbol{x})\}$:

$$g(\boldsymbol{\tau})\ f(\boldsymbol{x}) = g(\boldsymbol{\tau})\ (\boldsymbol{b}^T\boldsymbol{\Psi}(\boldsymbol{x})) = \boldsymbol{b}^T g(\boldsymbol{\tau})\ \boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{b}^T\boldsymbol{A}(\boldsymbol{\tau})\boldsymbol{\Psi}(\boldsymbol{x}) \tag{4.4}$$

and with steering functions given by $\boldsymbol{\alpha}(\boldsymbol{\tau}) = \boldsymbol{A}^T(\boldsymbol{\tau})\boldsymbol{b}$.

4.2.1 Construction of $1D$ equivariant function spaces for 1-parameter transformation groups

In our work we make use only of steerability under a 1-parameter transformation group, in particular, the rotation group with parameter being a rotation angle in the Fourier domain and acting on a $1D$ function $f(\phi)$. Thus, in the sequel, we will denote our basic coordinate by ϕ instead of $\boldsymbol{x} = x_1$. It can be shown that there exists always a change of coordinates such that any 1-parameter transformation group becomes a translation group in the new parameterization [28], thus, it is enough to consider the translation group with parameter ϕ. In our case, the rotation group, changing from cartesian coordinates to polar coordenates, reduces the problem to considering the translation group with parameter ϕ. The construction of equivariant function spaces is crucial in order to find functions which are steerable. The most elegant and simple way to construct equivariant function spaces is by using the concept of tangent space of a Lie group because it allows us to use simple linear Algebra as shown next.

Definition 18 *Given a 1-parameter trasformation group G which implies a coordinate transformation $\phi' = s_\tau(\phi)$, the infinitesimal generator of the transformation group L is defined by the differential operator:*

$$L = \frac{\partial x'}{\partial \tau} \frac{\partial}{\partial \phi} \bigg|_{\tau=0} \tag{4.5}$$

The tangent space Ω of the group G is defined as the set $\Omega = \{\tau L \mid \tau \in \mathbb{R}\}$ which can be viewed as a one-dimensional linear vector space with L being a one-dimensional basis vector. The crucial connection between the Lie group and the tangent space is that each element $g(\tau)$ in G is related to an element in Ω through an exponential map, namely:

$$g(\tau)\ f(\phi) = e^{\tau L} f(\phi) = \left(I + \tau L + \frac{1}{2!}\tau^2 L^2 + \cdots \right) f(\phi) \tag{4.6}$$

where τ is the parameter of the transformation group and $e^{\tau L}$ represents an infinite sum of differential operators (series expansion) [28, 107].

For the translation group, where the associated coordinate transformation is simply given by $\phi' = s_\tau(\phi) = \phi - \tau$, the infinitesimal generator is given by $L_t = -\frac{\partial}{\partial \phi}$ and the exponential map can be seen to be:

$$g_t(\tau)\ f(\phi) = e^{\tau L_t} f(\phi) = f(\phi) - \tau \frac{\partial f(\phi)}{\partial \phi} + \frac{1}{2!}\tau^2 \frac{\partial^2 f(\phi)}{\partial \phi^2} + \cdots = f(\phi - \tau) \tag{4.7}$$

The final result that gives a way to construct equivariant spaces [126] is given by the following Theorem [108, 109][3]:

Theorem 5 *The function space $V = span\{\psi_1(\phi), \ldots, \psi_J(\phi)\}$ is equivariant under the transformation group G if and only if V is closed under the action of the*

[3] This Theorem is still true for any k-parameter transformation group

infinitesimal generator L associated with the group G, that is, if and only if there is a matrix $\boldsymbol{B}$ such that:

$$L\boldsymbol{\Psi}(\phi) = \boldsymbol{B}\boldsymbol{\Psi}(\phi) \tag{4.8}$$

which is called the interpolation equation, and the interpolation matrix can be written as:

$$\boldsymbol{A}(\tau) = e^{\tau\boldsymbol{B}} = \boldsymbol{I} + \tau\boldsymbol{B} + \frac{1}{2!}\tau^2\boldsymbol{B}^2 + \cdots \tag{4.9}$$

Proof: See the proof in [108, 109].

If we apply Theorem 5 for the case of the translation group, we have an equivariant space V if and only if $L_t\boldsymbol{\Psi}(\phi) = \frac{\partial}{\partial\phi}\boldsymbol{\Psi}(\phi) = \boldsymbol{B}\boldsymbol{\Psi}(\phi)$ for a given $J \times J$ matrix. The solution to this ordinary differential vector equation is simply:

$$\boldsymbol{\Psi}(\phi) = e^{\phi\boldsymbol{B}}\boldsymbol{\Psi}(0) \tag{4.10}$$

where $\boldsymbol{\Psi}(0)$ (initial condition) is the column vector $\boldsymbol{\Psi}(\phi)$ evaluated at $\phi = 0$. Since the coordinates of the vector $\boldsymbol{\Psi}(0)$ can have any value, the general solution is a vector $\boldsymbol{\Psi}(\phi)$ such that $span\{\psi_1(\phi), \ldots, \psi_J(\phi)\} = R(e^{\phi\boldsymbol{B}})$ where $R(e^{\phi\boldsymbol{B}})$ means the column space of the matrix $e^{\phi\boldsymbol{B}} = \boldsymbol{I} + \phi\boldsymbol{B} + \frac{1}{2!}\phi^2\boldsymbol{B}^2 + \cdots$, which is the corresponding equivariant space. From the second part of Theorem 5, the interpolation matrix is given by $\boldsymbol{A} = e^{\tau\boldsymbol{B}}$ and the interpolation equation is $g_t(\tau)$ $\boldsymbol{\Psi}(\phi) = \boldsymbol{\Psi}(\phi - \tau) = e^{\tau\boldsymbol{B}}\boldsymbol{\Psi}(\phi)$.

Each *particular choice* for the matrix $\boldsymbol{B}$ gives rise to a particular set of basis functions, whose span is a particular equivariant space. Notice that once this matrix is chosen, the steering functions are simply readily obtained in a very simple way using (4.4). This simplicity in the derivation is the main advantage provided by the Lie Theory formulation as opposed to the derivations provided in [52, 100, 89, 90].

4.2.2 $2D$ Digital filter banks steerable under rotation constructed from equivariant spaces

In this section, we illustrate how the steerable filter designs provided by Freeman and Adelson [52] and Simoncelli [99] are actually a particular case of the formulation provided above based on Lie groups. Let $H(\omega_x, \omega_y)$ be the frequency response (DTFT) of a $2D$ digital filter $h(n_x, n_y)$ such that $H(\omega_x, \omega_y)$ is polar separable in the Fourier domain, that is, $H(\omega_x, \omega_y) = H(r, \phi) = B(r)\Theta(\phi)$ ($r = \sqrt{\omega_x^2 + \omega_y^2}$ and $\phi = \arctan(\omega_y/\omega_x)$ are the polar coordinates in the Fourier plane) , where the angular function $\Theta(\phi)$ gives the angular profile in the Fourier domain of the filter $h(n_x, n_y)$. The goal is to make the function $\Theta(\phi)$ steerable under the translation group so that $H(\omega_x, \omega_y)$ becomes steerable under the rotation group. The requirement of polar separability is not necessary, that is, the only necessary requirement is that for each r, the function $H(r, \phi)$ has to be steerable on the ϕ

coordinate. However, in this chapter, we concentrate in the particular case where $H(\omega_x, \omega_y)$ is polar separable, as in [52, 100].

Consider the particular choice of letting $\boldsymbol{B}$ be a $J \times J$ diagonal matrix with purely complex elements of conjugate pairs:

$$\boldsymbol{B} = \begin{pmatrix} j\lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & -j\lambda_1 & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & j\lambda_M & 0 \\ 0 & 0 & 0 & 0 & -j\lambda_M \end{pmatrix} \tag{4.11}$$

where $J = 2M$ and $\lambda_i \neq \lambda_k$, $i \neq k$. This results in an equivariant space given by the span of $2M$ linearly independent complex exponentials, that is, $V = span\{e^{j\lambda_1\phi}, e^{-j\lambda_1\phi},$
$\ldots, e^{j\lambda_M\phi}, e^{-j\lambda_M\phi}\}$. Let $\boldsymbol{\Psi}_o(\phi) = [e^{j\lambda_1\phi}, e^{-j\lambda_1\phi}, \ldots, e^{j\lambda_M\phi}, e^{-j\lambda_M\phi}]^T$. This is a valid set of basis functions for the equivariant space V. The corresponding interpolation matrix $\boldsymbol{A}_o(\tau)$ associated with $\boldsymbol{\Psi}_o(\phi)$ is given by $\boldsymbol{A}_o(\tau) = \text{diag}[e^{j\lambda_1\tau}, e^{-j\lambda_1\tau},$
$\ldots, e^{j\lambda_M\tau}, e^{-j\lambda_M\tau}]$. The final set of basis functions that is used to construct the

$2D$ filter bank is obtained performing a non-singular[4] linear transformation $\boldsymbol{P}$ on $\boldsymbol{\Psi}_o(\phi)$ as follows:

$$\boldsymbol{\Psi}(\phi) = \boldsymbol{P}\boldsymbol{\Psi}_o(\phi), \quad \boldsymbol{P} = \begin{pmatrix} \frac{\beta_1}{2}e^{-j\lambda_1\phi_1} & \frac{\beta_1}{2}e^{j\lambda_1\phi_1} & \dots & \frac{\beta_M}{2}e^{-j\lambda_M\phi_1} & \frac{\beta_M}{2}e^{j\lambda_M\phi_1} \\ \frac{\beta_1}{2}e^{-j\lambda_1\phi_2} & \frac{\beta_1}{2}e^{j\lambda_1\phi_2} & \dots & \frac{\beta_M}{2}e^{-j\lambda_M\phi_2} & \frac{\beta_M}{2}e^{j\lambda_M\phi_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\beta_1}{2}e^{-j\lambda_1\phi_J} & \frac{\beta_1}{2}e^{j\lambda_1\phi_J} & \dots & \frac{\beta_M}{2}e^{-j\lambda_M\phi_J} & \frac{\beta_M}{2}e^{j\lambda_M\phi_J} \end{pmatrix} \tag{4.12}$$

which gives $\psi_i(\phi) = \sum_{k=1}^{M} \beta_i \cos(\lambda_k(\phi - \phi_i))$. Now, taking the angular profile of the basic filter $H(\omega_x, \omega_y)$ to be $\Theta(\phi) = \sum_{k=1}^{M} \beta_k \cos(\lambda_k \phi)$, we have that $\psi_i(\phi) = \Theta(\phi - \phi_i)$, thus, in this particular case, the basis functions $\{\psi_i(\phi)\}_{i=1}^{J}$ are all obtained as different shifts of the angular function $\Theta(\phi)$, which is actually the function to be steered. The set of J angles $\{\phi_i\}_{i=1}^{J}$ are called basic angles or orientations. The interpolation matrix for $\boldsymbol{\Psi}(\phi)$ is readily given by:

$$\begin{gathered} g(\tau)\ \boldsymbol{\Psi}_o(\phi) = \boldsymbol{A}_o(\tau)\boldsymbol{\Psi}_o(\phi) \Leftrightarrow g(\tau)\ \boldsymbol{P}^{-1}\boldsymbol{\Psi}(\phi) = \boldsymbol{A}_o(\tau)\boldsymbol{P}^{-1}\boldsymbol{\Psi}_o(\phi) \\ \Leftrightarrow \boldsymbol{P}^{-1}(g(\tau)\ \boldsymbol{\Psi}(\phi)) = \boldsymbol{A}_o(\tau)\boldsymbol{P}^{-1}\boldsymbol{\Psi}(\phi) \Leftrightarrow g(\tau)\ \boldsymbol{\Psi}(\phi) = \boldsymbol{P}\boldsymbol{A}_o(\tau)\boldsymbol{P}^{-1}\boldsymbol{\Psi}(\phi) \end{gathered} \tag{4.13}$$

which means that $\boldsymbol{A}(\tau) = \boldsymbol{P}\boldsymbol{A}_o(\tau)\boldsymbol{P}^{-1}$, where $\boldsymbol{P}$ is given in (4.12). Thus, $\boldsymbol{A}_o(\tau)$ and $\boldsymbol{A}(\tau)$ are similar matrices. Since the function to be steered is $\Theta(\phi) =$

[4] Therefore, it is still a valid set of basis functions.

$\boldsymbol{b}^T\boldsymbol{\Psi}_o(\phi) = \boldsymbol{b}^T\boldsymbol{\Psi}_o(\phi)$, where $\boldsymbol{b} = [\frac{\beta_1}{2}, \frac{\beta_1}{2}, \ldots, \frac{\beta_J}{2}, \frac{\beta_J}{2}]^T$, using (4.4), the final set of steering functions $\boldsymbol{\alpha}(\tau) = [\alpha_1(\tau), \ldots, \alpha_J(\tau)]$ associated with $\boldsymbol{\Psi}(\phi)$ is given by

$$\boldsymbol{\alpha}^T(\tau) = \boldsymbol{b}^T\boldsymbol{P}^{-1}\boldsymbol{A}(\tau) = \boldsymbol{b}^T\boldsymbol{P}^{-1}\boldsymbol{P}\boldsymbol{A}_o(\tau)\boldsymbol{P}^{-1} = \boldsymbol{b}^T\boldsymbol{A}_o(\tau)\boldsymbol{P}^{-1} \tag{4.14}$$

satisfying that

$$g(\tau)\ \Theta(\phi) = \Theta(\phi - \tau) = \sum_{i=1}^{J} \alpha_i(\tau)\Theta(\phi - \phi_i) \quad \forall\tau \in \mathbb{R} \tag{4.15}$$

showing the following Lemma.

Lemma 4 *Given a set of basic angles $\{\phi_1, \ldots, \phi_J\}$, the set of steering functions that are needed to steer any angular profile of the type $\Theta(\phi) = \sum_{k=1}^{M} \beta_k \cos(\lambda_k\phi)$ is given by:*

$$\boldsymbol{\alpha}^T(\tau) = \boldsymbol{b}^T\boldsymbol{A}_o(\tau)\boldsymbol{P}^{-1} \tag{4.16}$$

where $\boldsymbol{b} = [\frac{\beta_1}{2}, \frac{\beta_1}{2}, \ldots, \frac{\beta_J}{2}, \frac{\beta_J}{2}]^T$, $\boldsymbol{A}_o(\tau)$ is the interpolation matrix associated with $\boldsymbol{\Psi}_o(\phi) = [e^{j\lambda_1\phi}, e^{-j\lambda_1\phi}, \ldots, e^{j\lambda_M\phi}, e^{-j\lambda_M\phi}]^T$ and $\boldsymbol{P}$ is given as in (4.12) and depends on the specific basic angles $\{\phi_1, \ldots, \phi_J\}$ that are chosen.

Proof: The proof is contained in the above explanation.

Let $H(\omega_x, \omega_y) = H(r, \phi)$ be, without loss of generality, the frequency response of a filter $h(n_x, n_y)$ oriented at 0 degrees. We can consider the set of J filters $\{h_{\phi_i}(n_x, n_y)\}_{i=1}^{J}$ oriented at angles $\{\phi_i\}_{i=1}^{J}$ having frequency responses:

$$H_{\phi_i}(\omega_x, \omega_y) = H_{\phi_i}(r, \phi) = H(r, \phi - \phi_i) = B(r)\psi_i(\phi) = B(r)\Theta(\phi - \phi_i) \quad (4.17)$$

where it can be seen that all the basic filters $\{H_{\phi_i}(\omega_x, \omega_y)\}_{i=1}^{J}$ are obtained as rotated versions of a unique filter $H(\omega_x, \omega_y)$. If an image $f(n_x, n_y)$ with DTFT $F(\omega_x, \omega_y)$ is filtered with these J filters, since convolution is a linear operation, it is possible to synthetize exactly the output of a filter oriented at an arbitrary orientation, by linearly combining (using the steering functions) the outputs of the J filters. Again, let $y(n_x, n_y)$ (with DTFT $Y(\omega_x, \omega_y)$) be the output of the filter $h(n_x, n_y)$ and $y_{\phi_i}(n_x, n_y)$ (with DTFT $Y_{\phi_i}(\omega_x, \omega_y)$) be the output of the i-th filter applied to the image $f(n_x, n_y)$. Freeman and Adelson [52] as well as Simoncelli [99] proposed to use angular profiles of the type $\Theta(\phi) = \cos^n(\phi)$ together with equispaced basic angles. This is actually a particular case of the above formulation because it is always posible to find a number $J = 2M$ of complex exponentials such that $\Theta(\phi) \in span\{e^{j\lambda_1\phi}, e^{-j\lambda_1\phi}, \ldots, e^{j\lambda_M\phi}, e^{-j\lambda_M\phi}\}$. Clearly, as n increases, the number of basis functions (basis filters) that are needed to steer $\Theta(\phi)$ is larger and therefore, the complexity for steering the filter output increases. On the other hand, when n increases, the support of the

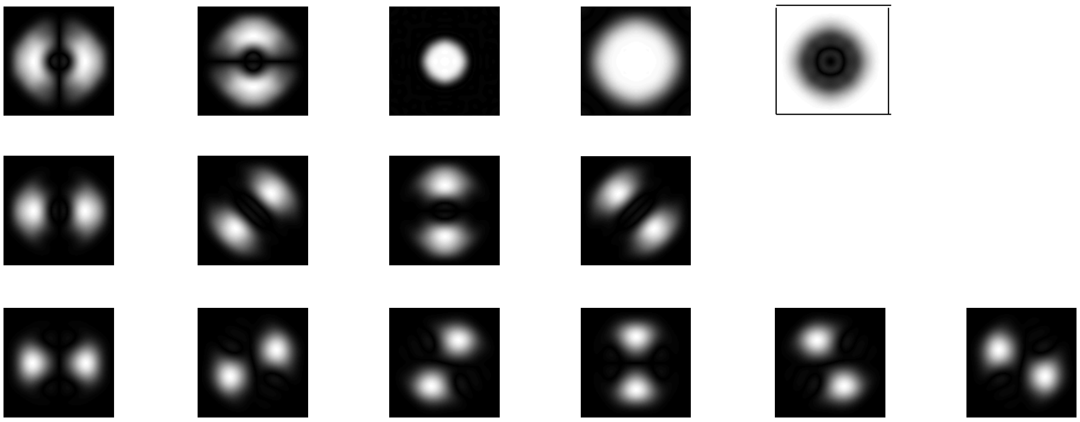


Figure 4.3: Firs row: $n = 1$, $\lambda_1 = 1$, $J = 2$. Low pass and High pass filters are also shown; Second row: $n = 3$, $\lambda_1 = 1$, $\lambda_2 = 3$, $J = 4$; Third row: $n = 5$, $\lambda_1 = 1$, $\lambda_2 = 3$, $\lambda_3 = 5$, $J = 6$

function $cos^n(\phi)$ decreases which implies that the angular bandwidth of the filter $H(\omega_x, \omega_y)$ becomes narrower (better resolution in angle). Fig. 4.3 illustrates the frequency responses for the filters corresponding to the cases of $n = 1, 3, 5$ where the basic angles have been chosen (as in [52, 99]) to be equiespaced starting with $\phi_1 = 0$ degrees, that is, $\phi_i = (i - 1)\pi/J$, $i = 1, \ldots, J$.

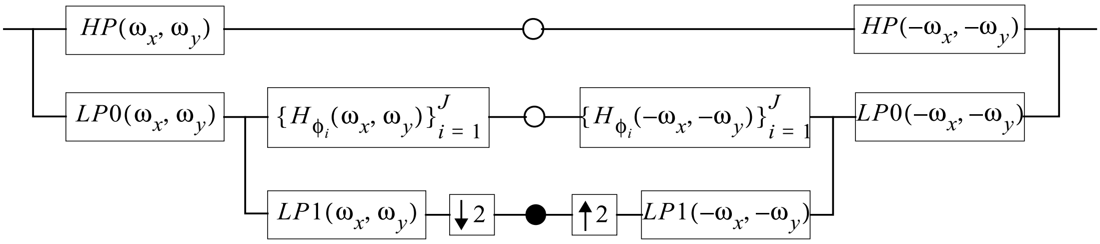

$HP(\omega_x, \omega_y)$
$HP(-\omega_x, -\omega_y)$
$LP0(\omega_x, \omega_y)$
$\{H_{\phi_i}(\omega_x, \omega_y)\}_{i=1}^{J}$
$\{H_{\phi_i}(-\omega_x, -\omega_y)\}_{i=1}^{J}$
$LP0(-\omega_x, -\omega_y)$
$LP1(\omega_x, \omega_y)$
↓2
↑2
$LP1(-\omega_x, -\omega_y)$

Figure 4.4: Structure of the steerable pyramid

In [52, 99, 98] it is shown how one can design a multi-scale, self-inverting, overcomplete pyramid decomposition (tight frame) in $\ell^2(\mathbb{Z}^2)$, where it is possible to perform steerability at every scale or resolution independently, that is, it is possible to steer any subband to any desired orientation. The pyramid structure can be seen in Fig. 4.4 where there is a total of 4 different filters to be designed, namely, $HP(\omega_x, \omega_y)$ (high pass filter), $LP0(\omega_x, \omega_y)$ (first low pass filter), $LP1(\omega_x, \omega_y)$ (second low pass filter) and the basic band pass filter $H(\omega_x, \omega_y)$, whose J rotated versions gives the set of J oriented band pass basis filters. Three constraints are optimized in the design of the J filters, namely: a) Aliasing cancellation for the filter $LP1(\omega_x, \omega_y)$, b) unity overall system response which is sufficient for perfect reconstruction since there is (approximately) no aliasing and c) recursive structure is inserted in the low pass branch of the second low pass filter $LP1(\omega_x, \omega_y)$. Since numerical methods are used for the optimization, the set of $2D$ digital filters do not satisfy exactly the flat power condition. The resulting transform is overcomplete by a factor of $\frac{4J}{3}$ approximately[5]:

$$\text{Redundancy} = J\left(1 + \frac{1}{4} + \frac{1}{16} + \cdots\right) = \frac{4J}{3} \qquad (4.18)$$

for the case of J basic band pass filters. For $J = 4$, we have a redundancy factor of approximately 5.3. Our work has not focused on improving the design of these filters and for more details about the design, see [99, 98]. However, some future work is considered in Chapter 5.

[5] Assuming that filtering in the pyramid is performed until there is only 1 pixel left

4.3 Angular oversampling in steerable transforms

Let $c(\boldsymbol{x}_o, \phi)$ represent the value of a transform coefficient corresponding to the output of a rotated filter with orientation ϕ for a certain spatial location $\boldsymbol{x}_o$. In a steerable transform with J basic orientations, at each scale or level, given the J coefficients $\{c(\boldsymbol{x}_o, \phi_1), c(\boldsymbol{x}_o, \phi_2), \ldots, c(\boldsymbol{x}_o, \phi_J)\}$, the transform coefficient $c(\boldsymbol{x}_o, \phi)$ for an angle (orientation) ϕ of that same spatial location will be given by:

$$c(\boldsymbol{x}_o, \phi) = \sum_{i=1}^{J} \alpha_i(\phi) c(\boldsymbol{x}_o, \phi_i) \quad \forall \phi \tag{4.19}$$

which gives a curve $c(\boldsymbol{x}_o \phi) \in W$, where $W = span\{\alpha_1(\phi), \alpha_2(\phi), \ldots, \alpha_J(\phi)\}$, which we call steerable curve. For angular profiles of the type $\Theta(\phi) = \cos^n(\phi)$, with n odd, notice that the frequencies $\lambda_1, \ldots, \lambda_M$ are all odd implying that the interpolation matrix $\boldsymbol{A}_o(\phi)$ satisfies that $\boldsymbol{A}_o(\phi + \pi) = -\boldsymbol{A}_o(\phi)$, and thus, from (4.14), it is clear that $c(\boldsymbol{x}_o, \phi + \pi) = -c(\boldsymbol{x}_o, \phi)$.

The important fact to observe is that if we have a set of transform coefficients $\{c(\boldsymbol{x}_o, \phi_i)\}_{i=1}^{K}$, where $K > J$, for a given location $\boldsymbol{x}_o$, all these points are constrained to belong to the same (deterministic) steerable curve (4.19), which is different for each spatial location $\boldsymbol{x}_o$ in the subband.

Since only J basic orientations are enough to interpolate exactly any other orientation, if we have $K > J$ quantization intervals, the steerability constraint induces a "consistency" constraint on these intervals which will help to reduce uncertainty in some of the quantization intervals. Equivalently, the fact that there is a deterministic curve linking the values of the coefficients at the different orientations implies that if we are given instead J quantization intervals $\{I(\boldsymbol{x}_o, \phi_1), \ldots, I(\boldsymbol{x}_o, \phi_J)\}$ at the J basic angles, there is a certain deterministic region $R(\boldsymbol{x}_o, \phi)$ of uncertainty at any other angle ϕ. We explain in Section 4.4.2 how to calculate the upper and lower bounds of these region.

Notice that using Lemma 4 in Section 4.2.2, we can calculate $c(\boldsymbol{x}_o, \phi)$ from any J angles (not necessarily equispaced) where the steering functions $\{\alpha_i(\phi)\}_{i=1}^J$ depend on the specific set of basic angles that are used $\{\phi_i\}_{i=1}^J$. In the same way, the region of uncertainty $R(\boldsymbol{x}_o, \phi)$ can be calculated making use of Lemma 4. As we will see later this has important implications in the quantization process.

We are interested in studying the case of having many orientations (oversteering) in the reprsentation, thus, we will generate, from the J basic orientations $\{\phi_i\}_{i=1}^J$, a certain number $K > J$ of subbands corresponding to the K orientations. We call this situation "angular oversampling" because there are more angles that are needed to perform steerability. This is because we want to study the trade-off between redundancy and accuracy in the representation assuming that a quantization of the coefficients takes place. The motivation for oversteering (e.g. increasing the number of orientations) is that when the number of oriented quantized subbands increases, we expect to be able to localize more efficiently energy that is oriented in angles which are different than the basic angles. This improved localization in energy, which consequently increases the coding capability, will try to compensate the corresponding increase in redundancy caused by the use of more orientations.

4.4 Techniques to represent efficiently the oversampled data

All the discussion that follows here concentrates on the problem of quantizing and coding all the oriented subbands (orientations) in only one scale, and more specifically it focuses only on the problem of removing redundancies across different orientations, not across scales. This particular coding problem is the one that can be solved efficiently by using (angular) consistency constraints. In addition to our proposed scheme, we could use a zero-tree based algorithm in order to remove statistical dependencies across the scales, as well as a context-based coding algorithm to remove redundancy inside each individual subband, all of them already developed in the context of wavelet transforms [118, 128, 97, 96, 27].

It is very important to note that in the light of a content based image retrieval application, since we are interested in extracting features from the transform domain, the distortion measure we consider is the average distortion of the transform coefficients.

It is clear that it is not efficient to code each oriented subband independently because of the large correlation among the different orientations in a given level. In the following subsections, we introduce two approaches to code the oversampled data making explicit use of the angular correlation among different subbands.

4.4.1 Energy localization in angle: Selection of maximums

An important fact we have observed is that angular oversampling permits to localize most of the energy of the image in a few coefficients (energy compaction in angle). This suggests that for a given spatial location, given a set of $K > J$ oriented subbands, one could use as the J basic angles those orientations having the largest energy, quantize these coefficients and perform predictive encoding of the rest of coefficients corresponding to orientations having lower energy. As an experiment, we generate, in a 1 level pyramid, from the 4 basic orientations, 10 and 100 equally spaced (in angle) different orientations from 0 to π (see Fig. 4.5). Then, we perform a thresholding as follows. For each number of orientations, we select at every spatial location $\boldsymbol{x}_o$, the maximum coefficient in magnitude $|c(\boldsymbol{x}_o, \phi_{max})|$ (ϕ_{max} indicates the angle for which the magnitude is maximum) out of all the orientations and set to 0 the rest of coefficients (corresponding to the other orientations). Then, we perform a simple thresholding in magnitude over the previously selected maximums so that $c(\boldsymbol{x}_o, \phi_{max}) \rightarrow 0$ if $|c(\boldsymbol{x}_o, \phi_{max})| < th$, where th is a preselected threshold. Finally, we reconstruct the original image ("Lena") from the resulting coefficients by using linear reconstruction convolving with all the steered filters. In Fig. 4.5, we see that we localize energy more efficiently when we increase the number of orientations, so there is a substantial gain in energy compaction. Notice that in this experiment we have used as distortion measure the distortion in the reconstructed image in order to illustrate the energy compaction. However, we are interested in taking advantage of this energy compaction property in the transform domain in order to code efficiently the K angular coefficients. This energy compaction in angle motivates the following very simple method to represent the data:

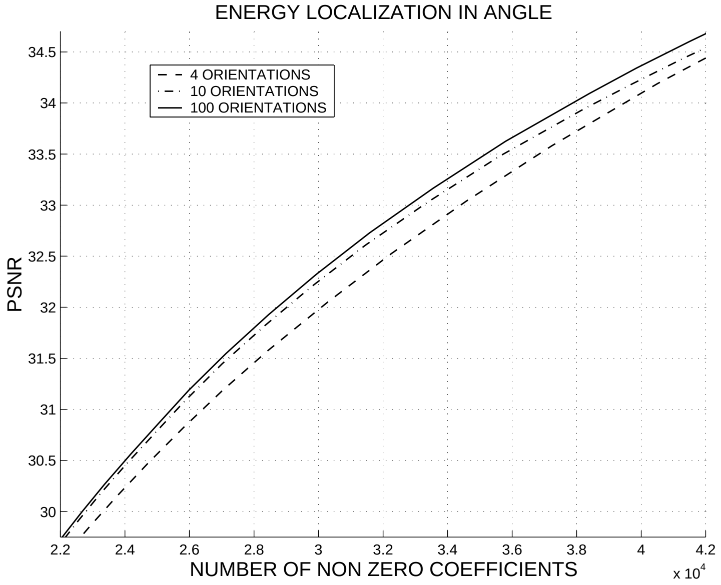


Figure 4.5: Energy compaction in angle. Steerable filter bank with $J = 4$ basic orientations. $PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right)$.

1. Let J be the number of necessary basic angles in the steerable represenation and $K > J$ be the number of available angles resulting in an oversteering. For every spatial location $\boldsymbol{x}_o$, depending on the coding accuracy we want to achieve, we select the T $(1 \leq T \leq J)$ angles for which the associated coefficients have largest magnitude out of the K available orientations and quantize their value according to some scalar quantizer whose stepsize will depend also on the desired coding resolution.

2. For every spatial location, using these T largest quantized values (assuming the other $J-T$ orientations having 0 coefficient values), we can estimate the steerable curve and therefore we can predict the values of the coefficients that correspond to all the other $K - T$ angles, and the prediction error is quantized.

This method, however, has the disadvantage that the coding resolution is only based on the increase in the energy localization as we increase the number of orientations, but it is not taking full advantage of the properties of smoothness of the steerable curve. It uses only T $(1 \leq T \leq J)$ to reduce the uncertainty in all the other $K-T$ angles. We have also observed that the prediction errors are very random, which means that it will not be possible to reduce the rate significantly even if entropy coding is used.

Notice that once some knowledge about the $K - T$ coefficients is obtained, it is possible (using different combinations of J angles) to reduce further the uncertainty of the values of the K coefficients. Thus, one should try to use all the available knowledge in order to reduce as much as possible the uncertainty produced after the quantization process. This is explored in the next section.

4.4.2 Use of Consistency Constraints

In this case, we establish two constraints, one due to the smoothness (steerability) constraints on the steerable curve linking all the angular coefficients of all the subbands (for every spatial location) and another one due to the quantization itself. As we increase the oversampling, we will have a reduction in the reconstruction errors (improving coding accuracy) which will try to compensate the increase in the bit rate.

We introduce two approaches to take advantage of the 2 consistency constraints for coding purposes.

4.4.2.1 Projection on convex sets (POCS)

The first strategy we provide is supported by POCS theory [131]. Consider that at a given spatial location $\boldsymbol{x}_o$, we apply a scalar quantizer Q (possibly with a different stepsize for each angle) to each of the angular coefficients, obtaining a

K-dimensional vector ($K > J$) $\boldsymbol{c}_q(\boldsymbol{x}_o) = [c_q(\boldsymbol{x_o}, \phi_1), \ldots, c_q(\boldsymbol{x_o}, \phi_K)]^T$ of quantized coefficients. Since the steerable curve $c(\boldsymbol{x}_o, \phi)$, originated from the unquantized coefficients, belongs to the space W spanned by the interpolation functions $\{\alpha_1(\phi), \ldots, \alpha_J(\phi)\}$, and we also know the quantization interval to which each coefficient belongs to, we can iterate projections on the following 2 sets:

1. S_1: The set of functions that belong to $W = span\{\alpha_1(\phi), \ldots, \alpha_J(\phi)\}$, which is a J-dimensional closed linear manifold (subspace) in $L^2(\mathbb{R})$.

2. S_2: The hypercube (contained in $\mathbb{R}^K$) defined by the K quantization intervals $\{I(\boldsymbol{x}_o, \phi_k)\}_{k=1}^{K}$. This hypercube is given by $Q^{-1}(\boldsymbol{c}_q(\boldsymbol{x}_o))$, that is, the set of vectors $\boldsymbol{c} \in \mathbb{R}^K$ such that $\boldsymbol{c}_q(\boldsymbol{x}_o) = Q(\boldsymbol{c})$.

The following Lemma allows us to use the Global convergence theorem of POCS [131] in order to find a final estimate $\hat{\mathbf{c}}(\boldsymbol{x}_o)$ which satisfy both constraints, that is, which belongs to $S_1 \cap S_2$.

Lemma 5 *The sets of constraints $S_1 = W$ and $S_2 = Q^{-1}(\boldsymbol{c}_q(\boldsymbol{x}_o))$ are convex and therefore, projecting alternatively on these 2 sets will converge to a K-dimensional vector $\hat{\mathbf{c}}(\boldsymbol{x}_o) \in S_1 \cap S_2$ (consistent estimate).*

Proof: See Appendix C.1.

Therefore, we propose to project iteratively on the convex sets S_1 and S_2.

The projection P_1 on S_1 is actually an orthogonal projection that takes the current estimate $\hat{\boldsymbol{c}}^{[j]}(\boldsymbol{x}_o)$ and updates it as follows:

$$\hat{\boldsymbol{c}}(\boldsymbol{x}_o) = \boldsymbol{P}_1\hat{\boldsymbol{c}}(\boldsymbol{x}_o) = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\hat{\boldsymbol{c}}(\boldsymbol{x}_o),\ \boldsymbol{A} = \begin{pmatrix} \alpha_1(\phi_1) & \alpha_2(\phi_1) & \cdots & \alpha_J(\phi_1) \\ \alpha_1(\phi_2) & \alpha_2(\phi_2) & \cdots & \alpha_J(\phi_2) \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_J(\phi_K) & \cdots & \cdots & \alpha_J(\phi_K) \end{pmatrix} \tag{4.20}$$

The initial estimate is just $\hat{\boldsymbol{c}}^{[0]}(\boldsymbol{x}_o) = \boldsymbol{c}_q(\boldsymbol{x}_o)$. Projection P_1 finds the vector in W that is closest to the current estimate $\hat{\boldsymbol{c}}^{[j]}(\boldsymbol{x}_o)$. Notice that since the K angles for which the steerable curve $c(\boldsymbol{x}_o, \phi)$ is sampled are fixed, the matrix $\boldsymbol{A}$ and hence the matrix $\boldsymbol{P}$, are known and fixed. The complexity of this projection increases with K since $\boldsymbol{P}$ is a $K \times K$ matrix, which means we need K^2 multiplications and $K(K-1)$ additions for each location $\boldsymbol{x}_o$.

The projection P_2 on S_2 is done as follows. Let $I(\boldsymbol{x}_o, \phi_k) = [L(\boldsymbol{x}_o, \phi_k), U(\boldsymbol{x}_o, \phi_k)]$. If $\hat{c}^{[j]}(\boldsymbol{x}_o, \phi_k) \in I(\boldsymbol{x}_o, \phi_k)$ (quantization interval at angle ϕ_k), $\hat{c}^{[j+1]}(\boldsymbol{x}_o, \phi_k) = \hat{c}^{[j]}(\boldsymbol{x}_o, \phi_k)$; if however, $\hat{c}^{[j]}(\boldsymbol{x}_o, \phi_k) \notin I(\boldsymbol{x}_o, \phi_k)$, we update the estimate to the bound of the quantization interval $I(\boldsymbol{x}_o, \phi_k)$ which is closest to $\hat{c}^{[j]}(\boldsymbol{x}_o, \phi_k)$, that is, if $|\hat{c}^{[j]}(\boldsymbol{x}_o, \phi_k) - L(\boldsymbol{x}_o, \phi_k)| < |\hat{c}^{[j]}(\boldsymbol{x}_o, \phi_k) - U(\boldsymbol{x}_o, \phi_k)|$, then, $\hat{c}^{[j+1]}(\boldsymbol{x}_o, \phi_k) = L(\boldsymbol{x}_o, \phi_k)$. Otherwise, $\hat{c}^{[j+1]}(\boldsymbol{x}_o, \phi_k) = U(\boldsymbol{x}_o, \phi_k)$. In order to perform this projection, we only need to perform k comparisons.

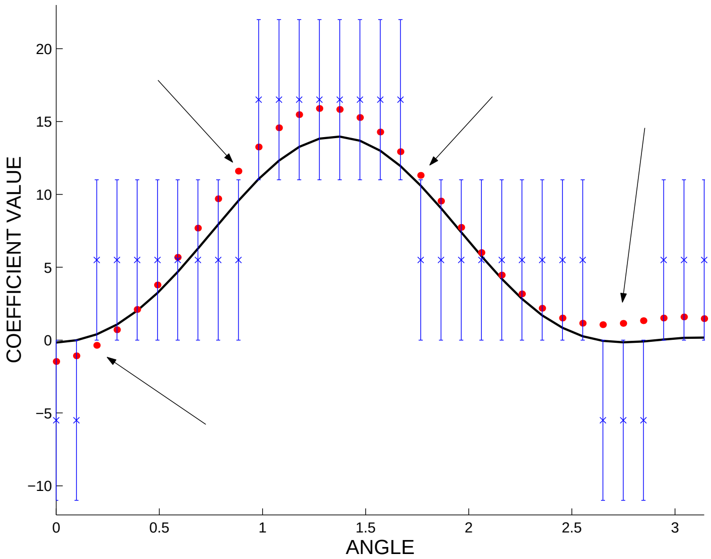


Figure 4.6: Example of non consistent angles. Number of angles = 32. Stepsize $\Delta = 11$. Original steerable curve (black curve) and estimated values (red dots) are shown where it can be seen that there are some angles for which the estimated values do not belong to the correct quantization intervals also shown, and hence, $\hat{\boldsymbol{c}}^{[j]}(\boldsymbol{x}_o) \notin Q^{-1}(\boldsymbol{c}_q(\boldsymbol{x}_o))$. The projection P_2 will be applied to those angles by moving the estimated values to the closest bounds of the correct quantization intervals.

Fig. 4.6 illustrates the case where this projection will be applied to some angles. The projection P_2 finds the vector in $Q^{-1}(\boldsymbol{c}_q(\boldsymbol{x}_o))$ that is closest to the current estimate $\hat{\boldsymbol{c}}^{[j]}(\boldsymbol{x}_o)$. Alternating projections means that we keep applying the composite projection $P = P_2P_1$. The property that ensures convergence is that a) every vector $\boldsymbol{c} \in S_1 \cap S_2$ is a fixed point of both P_1 and P_2, and hence of P and b) every fixed point of P is a vector in $S_1 \cap S_2$. Convexity of the sets is crucial in order to ensure that each projection is unique, that is, a projection of a vector in a non-covex set may not be unique.

The meaning of consistency here is that the estimate $\hat{\boldsymbol{c}}(\boldsymbol{x}_o)$ is consistent with all the knowledge available, that is, it is consistent with the quantization properties and it is consistent with the steerability property. Obviously, the accuracy of the estimate $\hat{\boldsymbol{c}}(\boldsymbol{x}_o)$ should increase when we increase the number of orientations K.

In practice, a finite number of alternating projections is used, namely, we keep projecting with P until the difference in norm of consecutive estimates becomes negligable. After a sufficient number of iterations, we then perform differential entropy coding, that is, we do entropy coding on the quantized differences (indices) between coefficients of adjacent angles. The total number of bits can be estimated by multiplying these entropies by the corresponding number of coefficients. This differential entropy coding is motivated from the fact that the steerable curve is always a smooth curve due to the fact that each of the steering functions $\alpha_i(\phi)$ is bandlimited and contains only a few harmonics.

This POCS-based technique presented here resembles that one used in [84, 111] in the context of A/D conversion of periodic bandlimited signals. In [84, 111], the alternating projections are applied in the time domain and the function space in that case is the space of periodic bandlimited signals. An estimate that belongs to the intersection of both convex sets is said to be a consistent estimate in [84, 111].

4.4.2.2 Regions of uncertainty

In the second strategy, the idea is, using J quantization intervals $\{I(\boldsymbol{x}_o, \phi_k)\}$ at J angles, we make use of the steerability proprety to constrain the region $R(\boldsymbol{x}_o, \phi)$ where all the $K - J$ coefficients should fall in (region of uncertainty). We can use any J angles, and each group of J angles will give rise to a different region of uncertainty. The intersection of all these regions of uncertainty will determine another region of uncertainty that will tend to be smaller due to the correlation among the different oriented subbands. We begin first quantizing the first J basic coefficients which were obtained from the steerable pyramid (usually, these are equispaced angles) and then the first region of uncertainty is determined. Each time a new orientation is added, we calculate other regions of uncertainty considering different sets of J angles. This procedure can be seen to be similar to the projection P_1 used in the POCS-based approach, which is actually a least squares linear fitting. The fitting with the interpolation functions $\{\alpha_i(\phi)\}_{i=1}^{J}$ tells us approximately where the original non quantized coefficients of the different angles should be. Similar information can be obtained by performing the intersections of all the different regions of uncertainty that come from the different groups of J angles that we have as we keep adding more quantization intervals at more angles, up until we have K quantization intervals and we have performed all the possible intersections. At this point, it is not possible to reduce more the uncertainty of the K angular coefficients $\boldsymbol{c}(\boldsymbol{x}_o)$. Finally, the reconstruction vector $\hat{\boldsymbol{c}}(\boldsymbol{x}_o)$ will be taken as the vector composed by the middle points of the final quantization intervals.

Notice that given a chosen number K of total orientations, we can precalculate easily the values of all the K sampled values of each of the interpolation functions $\{\alpha_i(\phi)\}$, since we know how to find all possible steering functions from Lemma 4. Once we have the corresponding samples of the steering functions, calculating each intersection requires $K \times J$ multiplications (in order to find the coefficient values at the K angles) and K comparisons for each spatial location $\boldsymbol{x}_o$. Next, we explain how to find the regions of uncertainty.

Let $R_U(\boldsymbol{x}_o, \phi)$ and $R_L(\boldsymbol{x}_o, \phi)$ denote the upper and lower limits of the region of uncertainty $R(\boldsymbol{x}_o, \phi)$. These upper and lower limits are curves themselves and the area between them on the domain $[0 \ \pi]$ will determine the total uncertainty. The problem of calculating $R(\boldsymbol{x}_o, \phi)$ can be stated as a linear programming problem for every angle ϕ, in the following way: given any J quantization intervals $\{I(\boldsymbol{x}_o, \phi_1), \ldots, I(\boldsymbol{x}_o, \phi_J)\}$ we have to find the values $\{c_1, \ldots, c_J\}$ which are the solution of the following two linear programming problems:

$$\begin{gathered} R_U(\boldsymbol{x}_o, \phi) = \max \textstyle\sum_{i=1}^{J} c_i \alpha_i(\phi), \qquad R_L(\boldsymbol{x}_o, \phi) = \min \textstyle\sum_{i=1}^{J} c_i \alpha_i(\phi) \\ \text{subject to: } l_i \leq c_i \leq u_i, \quad I(\boldsymbol{x}_o, \phi_i) = [l_i \, u_i], \quad i = 1, \ldots, J \end{gathered} \tag{4.21}$$

where $\Delta(\phi_i) = u_i - l_i$ is the stepsize of the scalar quantizer applied at ϕ_i and $\text{Width}(\boldsymbol{x}_o, \phi) = R_U(\boldsymbol{x}_o, \phi) - R_L(\boldsymbol{x}_o, \phi)$. We notice that these linear programming problems have bounded solutions because the set of constraints is bounded and the solutions $\{c_1, \ldots, c_J\}$ correspond to borders (upper or lower) of the quantization intervals $\{I(\boldsymbol{x}_o, \phi_1), \ldots, I(\boldsymbol{x}_o, \phi_J)\}$. The following theorem gives a complete characterization of the region of uncertainty $R(\boldsymbol{x}_o, \phi)$.

Theorem 6 *Given J quantization intervals $\{I(\boldsymbol{x}_o, \phi_1), \ldots, I(\boldsymbol{x}_o, \phi_J)\}$, the region of uncertainty $R(\boldsymbol{x}_o, \phi)$ is determined by:*

$$\begin{aligned} R_U(\boldsymbol{x}_o, \phi) &= \sum_{i=1}^{J} U(\phi_i)\alpha_i(\phi) \quad \textit{where} \quad U(\phi_i) = \left\{ \begin{array}{ll} u_i & \textit{if } \alpha_i(\phi) > 0 \\ l_i & \textit{if } \alpha_i(\phi) < 0 \end{array} \right\} \\ R_L(\boldsymbol{x}_o, \phi) &= \sum_{i=1}^{J} L(\phi_i)\alpha_i(\phi) \quad \textit{where} \quad L(\phi_i) = \left\{ \begin{array}{ll} l_i & \textit{if } \alpha_i(\phi) > 0 \\ u_i & \textit{if } \alpha_i(\phi) < 0 \end{array} \right\} \end{aligned}$$

where $I(\boldsymbol{x}_o, \phi_i) = [l_i\, u_i]$. Moreover, the following properties are satisfied:

1. *Width$(\boldsymbol{x}_o, \phi) = R_U(\boldsymbol{x}_o, \phi) - R_L(\boldsymbol{x}_o, \phi) = \sum_{i=1}^{J} \Delta(\phi_i)|\alpha_i(\phi)|$, where $\Delta(\phi_i) = u_i - l_i$, $i = 1, \ldots, J$.*

2. *The central curve $\hat{c}(\boldsymbol{x}_o, \phi)$ of the resulting region of uncertainty $R(\boldsymbol{x}_o, \phi)$ is a steerable curve given by $\hat{c}(\boldsymbol{x}_o, \phi) = \sum_{i=1}^{J} \hat{c}(\boldsymbol{x}_o, \phi_i)\alpha_i(\phi)$ where $\hat{c}(\boldsymbol{x}_o, \phi_i) = (l_i + u_i)/2$, $i = 1, \ldots, J$.*

3. $R(\boldsymbol{x}_o, \phi) = R_0(\boldsymbol{x}_o, \phi) + \hat{c}(\boldsymbol{x}_o, \phi)$*, where* $R_0(\boldsymbol{x}_o, \phi)$ *is the region of uncertainty with bounds given by* $R_{U_0}(\boldsymbol{x}_o, \phi) = \sum_{i=1}^{J} \left(\frac{\Delta(\phi_i)}{2}\right) |\alpha_i(\phi)|$ *and* $R_{L_0}(\boldsymbol{x}_o, \phi) = -R_{U_0}(\boldsymbol{x}_o, \phi)$*.*

Proof: See Appendix C.2. Therefore, Width$(\boldsymbol{x}_o, \phi)$ does not depend on the relative position of the quantization interval centers but it only depends on the J basic angles that are used which determine the set of steering functions $\{\alpha_i(\phi)\}_{i=1}^{J}$ and on the quantization intervals $\Delta(\phi_i) = u_i - l_i$, $i = 1, \ldots, J$ that are used for the basic angles. Notice that this theorem implies that the linear programming problem can be normalized to the case where all the quantization intervals are centered at 0. Figs. 4.7(a) and 4.7(b) show the region of uncertainty and the width of this region for a particular case where the basic quantization intervals have an stepsize $\Delta = 5$ and the basic angles are 0,$\frac{\pi}{4}$,$\frac{\pi}{2}$ and $\frac{3\pi}{4}$. It can be seen

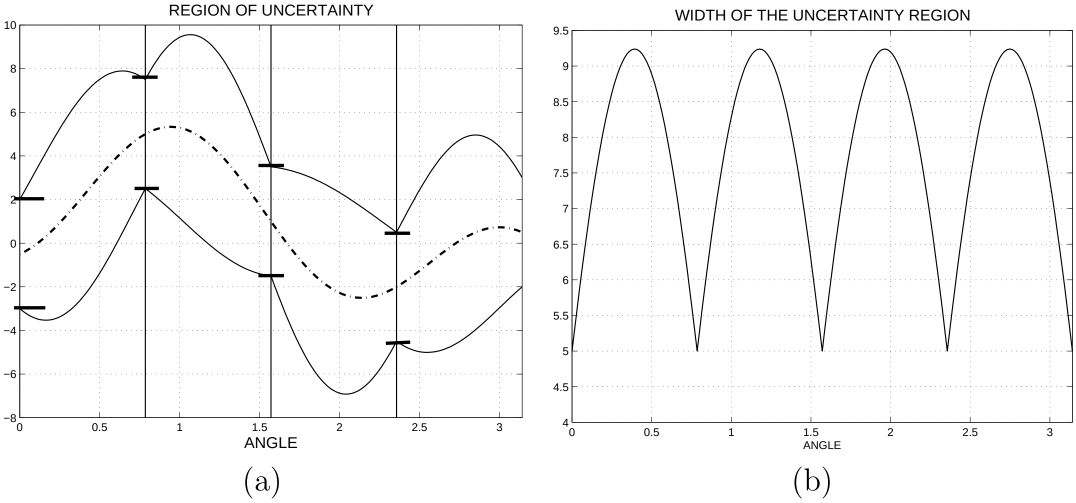


Figure 4.7: (a) Region of uncertainty $R(\boldsymbol{x}_o, \phi)$; (b) Width$(\boldsymbol{x}_o, \phi)$.

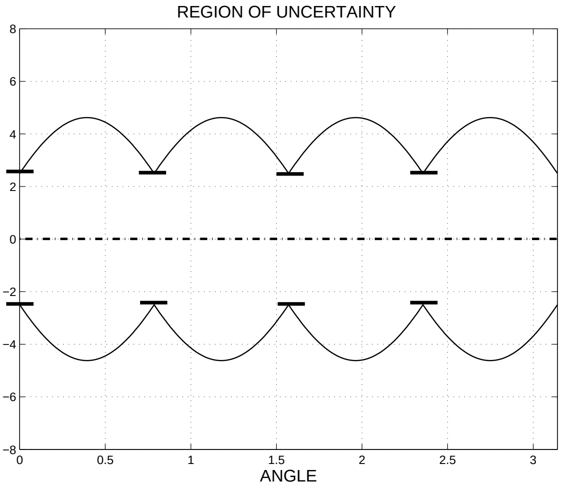


Figure 4.8: Normalization of the problem to the case where all the quantization intervals are centered at 0.

in these figures that the width of the uncertainty Width$(\boldsymbol{x}_o, \phi)$ increases as we move away from the basic angles, as expected. Fig. 4.8 shows the normalization of the problem. Since each different set of J basic angles gives rise to a different region of uncertainty, when we have more than J angles, we can always consider taking the intersection of different regions of uncertainty. An illustrative example of intersecting regions is given in Fig. 4.9 where there is a total of $K = 6$ angles and the intersection of 3 regions of uncertainty is shown.

4.4.3 Experimental Results

We perform experiments with the "Lena" image and more particularly, we study the coding performance of our algorithms on each level of the steerable pyramid. Similar results have been found in each of the levels and we show here, as a good

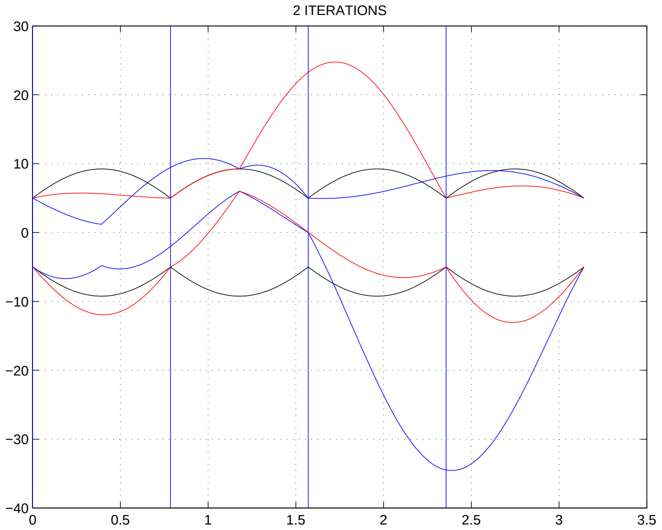
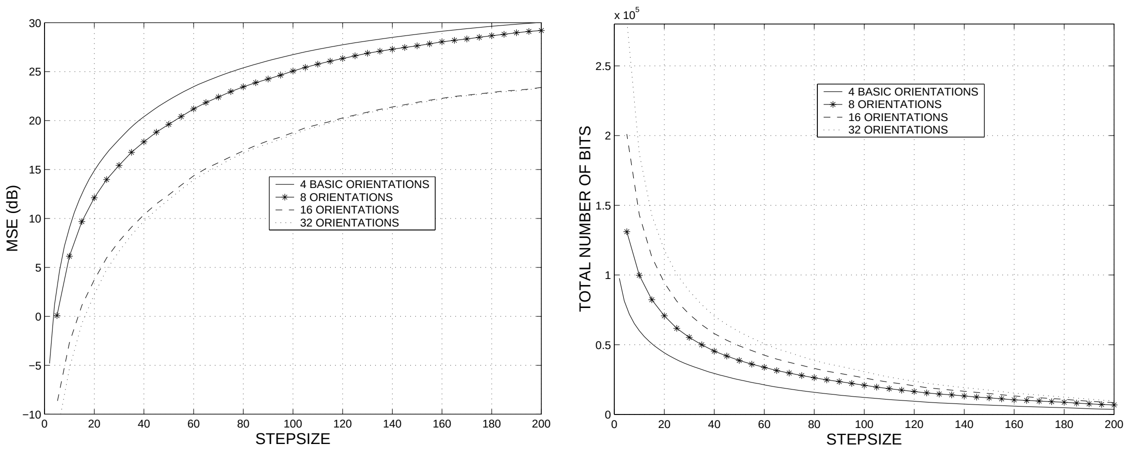


Figure 4.9: An example with $K = 6$ angles (2 additional angles) and 3 regions of uncertainty. The intersection is reducing the uncertainty.

representative, the 3rd level of the steerable pyramid. In our experiments, we use a steerable transform with $J = 4$ basic angles and we oversteer the representation to 8, 16 and 32 orientations. We compare the coding performance between the non oversteered case, that is, using only 4 basic angles (with direct quantization) chosen equally spaced angles as $0, \frac{\pi}{4}, \frac{\pi}{2}$ and $\frac{3\pi}{4}$, and the 3 other cases with oversteering. The comparison has been made in terms of the total number of bits which is measured as explained in Section 4.4.2 and the MSE is averaged over all the spatial locations and over all the angles (not only the initial basic J angles) that we have in each case and thus, it is measured on the transformed domain [6]. In order to get a range of values for the MSE, we have changed the value of the stepsize Δ for the scalar quantizer which is applied initially at every angle. We

[6] Notice that we are considering that the MSE is measured over the K angles.



(a) (b)

Figure 4.10: (a) MSE vs stepsize; (b) Bits vs stepsize.

assume that initially, the same stepsize is applied to all the angles. We have used the method based on POCS and in our future work we expect to use the second method based on intersection of regions of uncertainty. Fig. 4.10(a) shows how the MSE is reduced as we increase the angular oversampling in the representation. From 4 orientations to 8 and 16 orientations, the improvements are 2.5 dB and 6 dB approximately, respectively. We also notice that for 32 orientations, the MSE is almost not reduced with respect to 16 orientations. The interpretation of this is that the smoothness of the steerable curve $c(\boldsymbol{x}_o, \phi)$, which in this particular case contains 2 harmonics ($\lambda_1 = 1$, $\lambda_2 = 3$) is so strong that using more constraints (more angles) is useless. In Fig. 4.11, we can see that there is a gain for low rates in the case of using 16 and 32 orientations with respect to using only the 4 basic orientations. This is because for these low rates, the reduction in

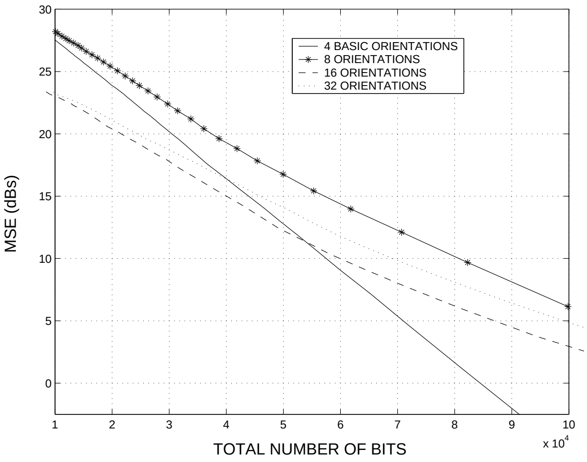


Figure 4.11: MSE vs the total number of bits.

MSE as we increase the number of orientations is faster than the corresponding increase in the number of bits, resulting in a coding gain.

4.5 Rotation Invariance in Content-based image retrieval

As mentioned in Section 4.1, one of the main drawbacks of using critically sampled wavelet filter banks is their inability to provide rotation invariance when they are applied in texture recognition and retrieval applications. In order to illustrate this problem, consider the following examples. Fig. 4.12 shows the subbands obtained using a steerable pyramid with 4 basic angles and with 3 levels when the input image is the texture "wood" from the Brodatz set [85] rotated (counter-clockwise) 30 degrees. It can be seen observed that the 2 subbands with more

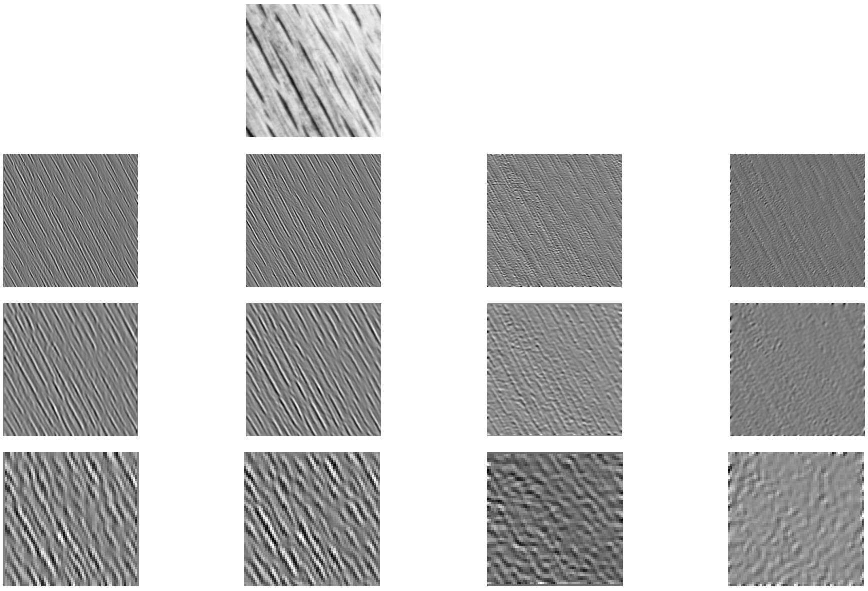


Figure 4.12: Texture "wood" rotated 30 degrees. Subbands obtained from the steerable pyramid with 4 basic angles with 3 levels. Subbands have been scaled to have the same size. The rows represent the scales, from top (coarsest) to bottom (finest). The columns represent the orientations, from left to right: 0, 45, 90, 135 degrees.

energy are the subband oriented at 45 degrees (largest energy) followed by the subband oriented at 0 degrees. Fig. 4.13 shows the same steerable pyramid, now with the input image being the same texture "wood" rotated 150 degrees. In this case, the 2 subbands with more energy are the subband oriented at 135 degrees (largest energy) followed by the subband oriented at 0 degrees. Notice that if we consider the sequence of angles 0, 45, 90, 135, 180, 225, ... and take into account that $c(\boldsymbol{x}_o, \phi + \pi) = -c(\boldsymbol{x}_o, \phi)$, what is observed is that the energy has gone from the angles $\{0, 45\}$ in Fig. 4.12 to the angles $\{135, 180\} \simeq \{0, 135\}$ in Fig. 4.13, where 180 degrees can be identified as 0. This is exactly the property

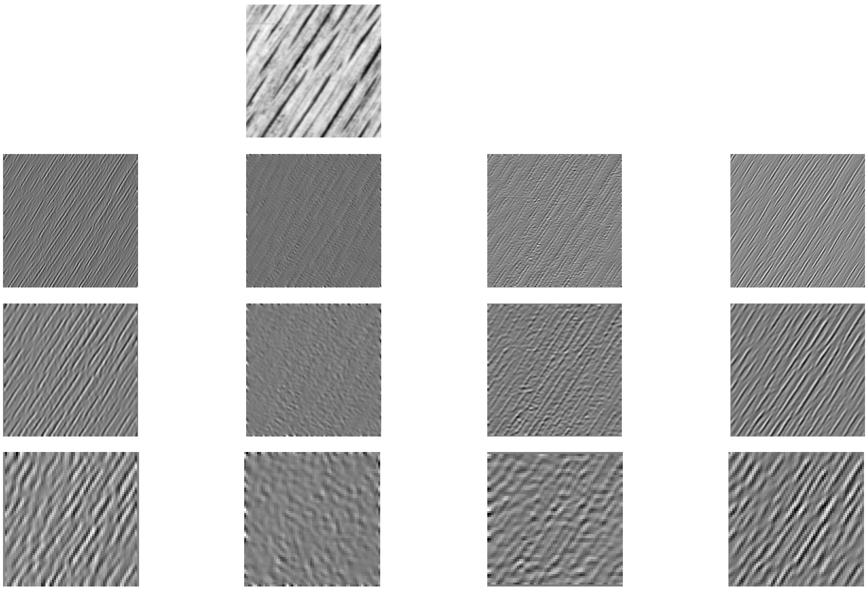


Figure 4.13: Texture "wood" rotated 150 degrees. Subbands obtained from the steerable pyramid with 4 basic angles with 3 levels. Subbands have been scaled to have the same size. Same order as in Fig. 4.12.

that identifies rotation invariance, that is, a rotation in the image, corresponds to a shift across the orientations in the steerable pyramid. As it is shown later, this is a very useful quality of the steerable representation in order to identify textures samples which are rotated versions of the same texture.

	$\phi_1 = 0$	$\phi_2 = 45$	$\phi_3 = 90$	$\phi_4 = 135$
Level 1	0.92	1.58	0.09	0.07
Level 2	8.47	13.61	0.66	0.49
Level 3	27.11	41.27	3.17	2.54

Table 4.1: Steerable pyramid for "wood" rotated at 30 degrees: Percentages (%) of total energy in the different subbands.

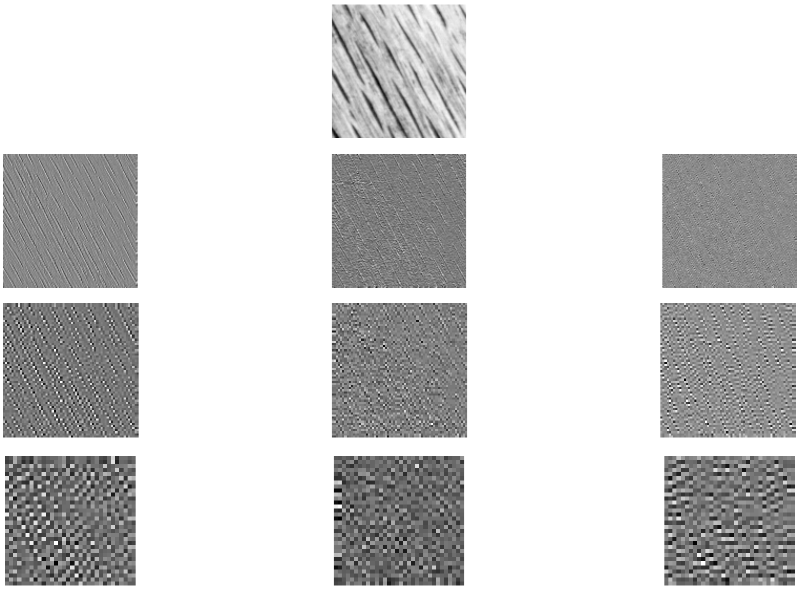


Figure 4.14: Texture “wood” rotated 30 degrees. Subbands obtained from a wavelet pyramid with 3 levels using the ’daub3’ filter bank. Subbands have been scaled to have the same size. The rows represent the scales, from top (coarsest) to bottom (finest). First column = 0, Second column = 90 degrees, Third column = 45 and 135 degrees together.

On the other hand, Fig. 4.14 shows the subbands obtained using the ’daub3’ filter bank with a 3-level pyramid when the input image is “wood” at 30 degrees and Fig. 4.15 shows the same wavelet pyramid, now with the input image being “wood” at rotated at 150 degrees. In this case, the previous behaviour is not observed. First, since in this wavelet transform there is no capability of distinguishing between having energy at 45 degrees and having energy at 135 degrees, the subband corresponding to the diagonals (45 and 135 degrees together), has approximately the same amount of energy in Fig. 4.14 and in Fig. 4.15. It is not possible to identify a shift across the columns from Fig. 4.14 to Fig. 4.15,

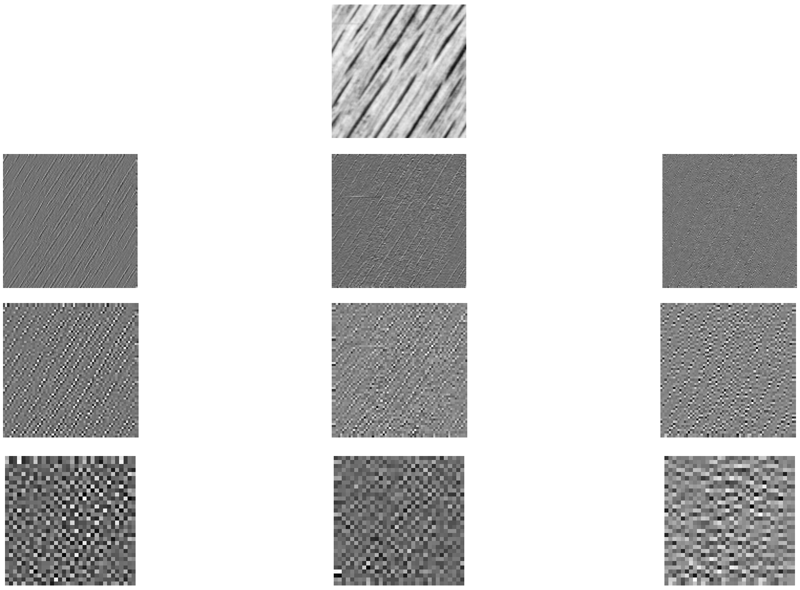


Figure 4.15: Texture “wood” rotated 150 degrees. Subbands obtained from a wavelet pyramid with 3 levels using the ’daub3’ filter bank. Subbands have been scaled to have the same size. Same order as in Fig. 4.14.

and when there is a rotation in the image, the energy is spread out over all the orientations. Tables 4.1 and 4.2 give the percentages of the total energy that is contained in each subband for the case of a steerable pyramid with 3 levels and $J = 4$ basic orientations. It can be seen that most of the energy is concentrated in the third level and that the energy moves to the right with a circular shift

	$\phi_1 = 0$	$\phi_2 = 45$	$\phi_3 = 90$	$\phi_4 = 135$
Level 1	0.77	0.07	0.14	1.89
Level 2	6.70	0.47	0.94	15.22
Level 3	23.18	2.38	3.89	44.32

Table 4.2: Steerable pyramid for “wood” rotated at 150 degrees: Percentages (%) of total energy in the different subbands.

	$\phi_1 = 0$	$\phi_2 = 90$	$\phi_3 = 45, 135$
Level 1	0.53	0.08	0.04
Level 2	10.31	0.76	1.65
Level 3	61.65	4.82	20.15

Table 4.3: Wavelet pyramid for "wood" rotated at 30 degrees: Percentages (%) of total energy in the different subbands.

	$\phi_1 = 0$	$\phi_2 = 90$	$\phi_3 = 45, 135$
Level 1	0.45	0.1087	0.05
Level 2	9.11	1.2390	2.47
Level 3	50.82	7.0284	28.69

Table 4.4: Wavelet pyramid for "wood" rotated at 150 degrees: Percentages (%) of total energy in the different subbands.

as explained previously. Tables 4.3 and 4.4 show the percentages of total energy for the case of a typical wavelet pyramid with 3 levels. As it is observed, the amounts of energy in the third level are similar in both cases and the energies are also higher in both cases for the orientations corresponding to 0 degrees and $45, 135$ degrees.

This makes it very difficult to be able to recognize rotated versions of the same texture.

4.5.1 Content-based image retrieval architecture

We consider a very simple architecture for a content-based image retrieval system, as shown in Fig. 4.16. This is a very standard approach which is typically used

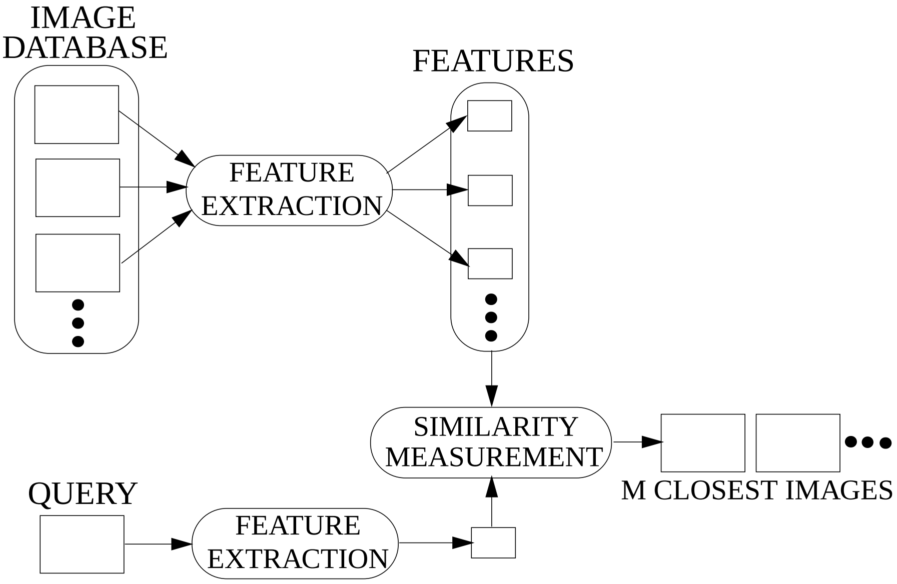


Figure 4.16: Image retrieval system architecture

in practice. The 2 most important parts of this system are: a) feature extraction, where a set of features, usually called image signatures, is generated in order to represent as good as possible the content of each image in the database. The number of features is much smaller than the size of the image or the subbands obtained from a multiresolution decomposition (e.g., wavelet pyramid or steerable pyramid); b) similarity measurement, where a distance between the query image and each image in the database is computed. The M images with the smallest distance will be retrieved. Low level features are used in these systems, such as color, texture, shape, etc.. Our focus is on using texture information for image retrieval with rotation invariance. Thus, in our work, the input to our system will be subblocks (containing texture information) of a larger image. Therefore, in a real application, a segmentation algorithm would have to be used in conjunction with this system.

Our goal is to construct an image retrieval system which can recognize the situation where the query image is a rotated version of some image already present in the database. Our database is organized in such a way that for each "type" of texture, we have a set of texture samples (image subblocks), all of them obtained from the same (larger) texture image oriented at a fixed orientation (e.g. 0 degrees). Therefore, if the query image is already in the image database, we want to maximize the percentage of textures (from the database) in the M closest texture samples which are rotated versions of the query image.

In the next two sections we explain in detail the feature extraction and the similarity measurement processes.

4.5.2 Feature extraction

Since we are interested in achieving rotation invariance, the feature extraction we consider is based on the subbands obtained from a steerable pyramid. In this context, it is important to use features which are as "steerable" as possible, that is, given the features of a texture oriented at an angle ϕ, the features corresponding to the same texture but oriented at an angle ϕ', can be approximately estimated from the features at angle ϕ without actually having to calculate the features for the rotated version. This is a crucial property that will be used in the similarity measurement. In this work, we try to achieve a good performance using simple energy-based features.

Consider the problem of calculating the average energy $E^l(\phi)$ of a subband oriented at an arbitrary angle ϕ in a level l, that is, $E^l(\phi) = \left(\frac{1}{N_l}\right) \sum_{k=1}^{N_l} (c^l(\boldsymbol{x}_k, \phi))^2$, where N_l is the number of pixels of each of the subbands in level l and $c^l(\boldsymbol{x}_k, \phi)$ is the value of the transform coefficient at angle ϕ, location $\boldsymbol{x}_k$ and level l. It is very simple to show that $E^l(\phi)$ can be calculated from the energies (sampled autocorrelations) of the basic J subbands and all the sampled cross-correlations between each pair of basic subbands.

Fact 3 *Given a steerable pyramid with J basic angles $\{\phi_1, \ldots, \phi_J\}$ and L levels, the function $E^l(\phi)$ is given by:*

$$E^l(\phi) = \boldsymbol{\alpha}^T(\phi)\boldsymbol{C}^l\boldsymbol{\alpha}(\phi), \quad \boldsymbol{\alpha}(\phi) = \begin{pmatrix} \alpha_1(\phi) \\ \vdots \\ \alpha_J(\phi) \end{pmatrix}, \quad \boldsymbol{C}^l = \begin{pmatrix} C^l_{11} & C^l_{12} & \cdots & C^l_{1J} \\ C^l_{21} & C^l_{22} & \cdots & C^l_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ C^l_{J1} & \cdots & \cdots & C^l_{JJ} \end{pmatrix} \tag{4.22}$$

where $C^l_{ij} = \left(\frac{1}{N_l}\right) \sum_{k=1}^{N_l} c^l(\boldsymbol{x}_k, \phi_i) c^l(\boldsymbol{x}_k, \phi_j) = C^l_{ji}$, $l = 1, \ldots, L$.

Proof: See Appendix C.3.

Notice that each diagonal element of $\boldsymbol{C}^l$ correspond to $C^l_{ii} = E^l(\phi_i)$, that is, the average energy at the basic angle ϕ_i, while the off-diagonal elements correspond to sampled cross-correlations between the subbands corresponding to each pair of basic angles. Since the basic (steerable) filters are almost pure bandpass filters, that is, since they have approximately zero mean, the subbands obtained by convolution with these filters will have also approximately zero mean. As a consequence, the sampled correlation matrix $\boldsymbol{C}^l$ will be approximately equal to the sampled covariance matrix. Notice that since $c(\boldsymbol{x}_o, \phi + \pi) = -c(\boldsymbol{x}_o, \phi)$, clearly, $E^l(\phi + \pi) = E^l(\phi)$, that is, $E^l(\phi)$ is a periodic function with period equal to π.

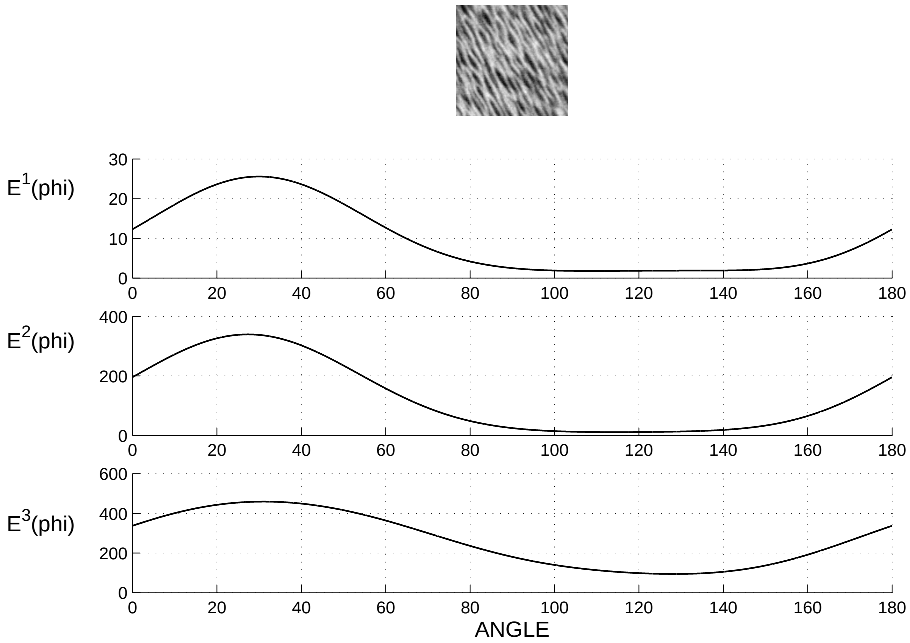


Figure 4.17: Energy profile $E^l(\phi)$, $l = 1, 2, 3$, for “water” at 30 degrees

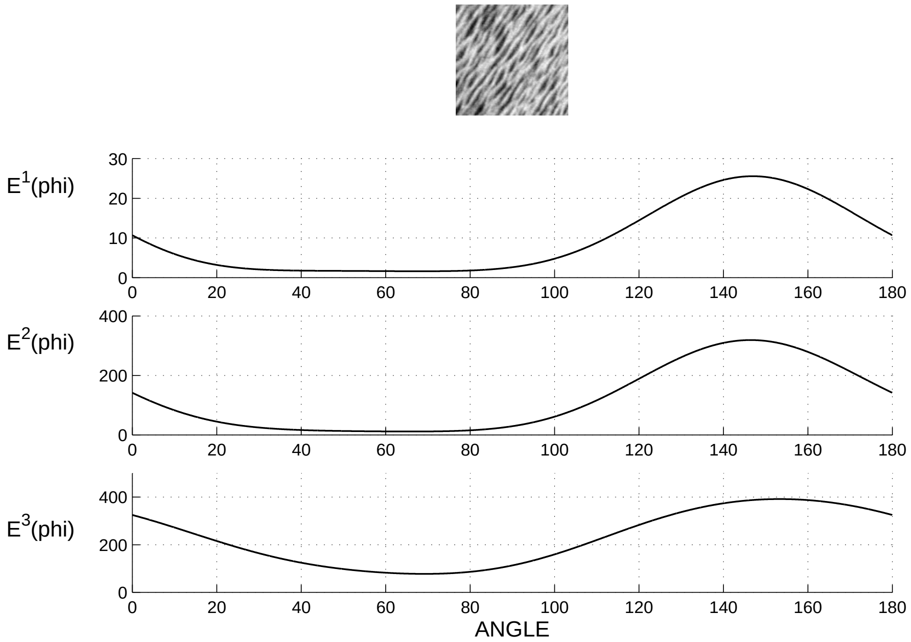


Figure 4.18: Energy profile $E^l(\phi)$, $l = 1, 2, 3$, for “water” at 150 degrees

Given a perfectly homogeneous texture I with energy profile $E^l_I(\phi)$ at level l, if this image is rotated counter-clockwise by an angle θ, obtaining an texture I_θ, then, we will have that[7] $E^l_{I_\theta}(\phi) = E^l_I(\phi - \theta)$, that is, a rotation of an texture corresponds to a shifted version of the energy profile. This can be observed comparing the profiles of energy shown in Fig. 4.17 and Fig. 4.18.

These arguments motivate the use of the correlation matrices $\{\boldsymbol{C}^l\}_{l=1}^L$ as features in our system. Notice that since each matrix $\boldsymbol{C}^l$ is symmetric, the total number of features will be $J(J+1)L/2$. Thus, the interdependencies between

[7] Even if the texture is perfectly homogeneous, the effect of the borders and the fact that we have a cubic grid (not continuous) will yield some deviation from an exact shift of the curve $E^l_I(\phi)$.

different orientations in terms of cross-correlations are necessary in order to characterize the energy profile of an arbitrary rotation of a given texture. We do not consider the use of the energy of the low-pass residual subband as a feature. Obviously, as J (number of basic orientations) increases, the resolution in angle increases and the energy profile $E^l(\phi)$ will be more accurate.

4.5.3 Similarity Measurement

In the similarity measurement, we are interested in making use of the steerability property in order to be able to align the features corresponding to rotated versions of the same texture, that is, the steerability property should be used to identify equivalent features, where equivalency will correspond to having different rotated versions of a unique texture.

The next proposition shows that the sampled correlation matrix $\boldsymbol{C}_I^l$ for a texture[8] at a given level and the sampled correlation matrix $\boldsymbol{C}_{I_\theta}^l$ for the same texture but rotated counter-clockwise by an angle θ, are related in a simple way.

[8] Here, we also assume that the texture is perfectly homogeneous and continuous

Proposition 2 *Given a steerable representation with J basic angles, the correlation matrices $\boldsymbol{C}^l_{I_\theta}$ and $\boldsymbol{C}^l_I$, both evaluated with respect to the same set of basic angles $\{\phi_1, \ldots, \phi_J\}$, are related as follows:*

$$\boldsymbol{C}^l_{I_\theta} = \boldsymbol{R}(\theta)\boldsymbol{C}^l_I\boldsymbol{R}^T(\theta), \quad \boldsymbol{R}(\theta) = \begin{pmatrix} \alpha_1(\phi_1-\theta) & \alpha_2(\phi_1-\theta) & \cdots & \alpha_J(\phi_1-\theta) \\ \alpha_1(\phi_2-\theta) & \alpha_2(\phi_2-\theta) & \cdots & \alpha_J(\phi_2-\theta) \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1(\phi_J-\theta) & \alpha_2(\phi_J-\theta) & \cdots & \alpha_J(\phi_J-\theta) \end{pmatrix} \tag{4.23}$$

In the particular case where the J basic angles are taken to be equiespaced, then $\boldsymbol{R}(\theta)$ becomes an orthogonal matrix for any θ, and therefore, $\boldsymbol{C}^l_{I_\theta}$ and $\boldsymbol{C}^l_I$ become orthogonally equivalent.

Proof: See Appendix C.4.

This property holds for every level independently. However, as shown in Fig. 4.17 and Fig. 4.18, when a texture is rotated, all the decomposition levels, will be equally rotated. This means that given a texture I and a rotated version I_θ of it, the Frobenius norms $\|\boldsymbol{C}^l_I - \boldsymbol{R}(-\theta)\boldsymbol{C}^l_{I_\theta}\boldsymbol{R}^T(-\theta)\|_F$ (same rotation angle for all the levels), $l = 1, \ldots, L$, will tend to be small.

Taking all this into account, the similarity measurement $D(I_1, I_2)$ between 2 different textures I_1 and I_2 that we use is the following:

$$D(I_1, I_2) = \mathrm{Min}_{\theta} \left(\sum_{l=1}^{L} \| \boldsymbol{C}^l_{I_1} - \boldsymbol{R}(-\theta) \boldsymbol{C}^l_{I_2} \boldsymbol{R}^T(-\theta) \|_F \right) \tag{4.24}$$

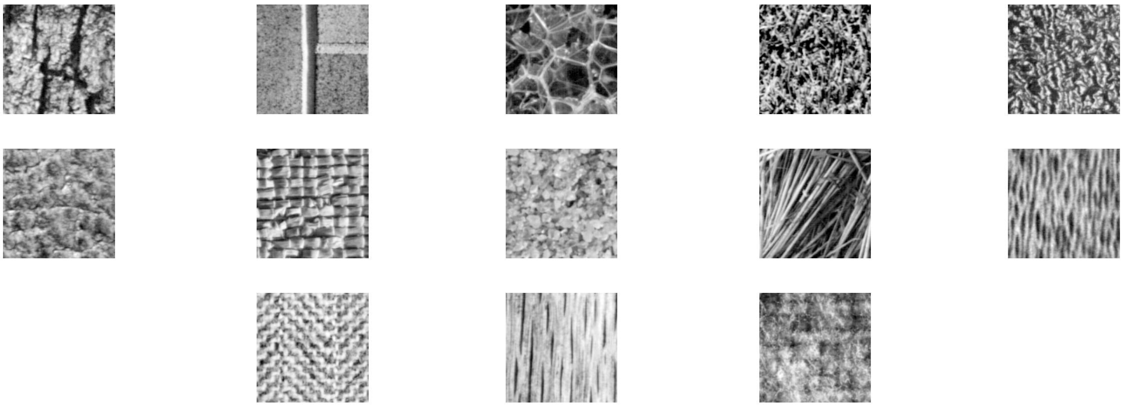


Figure 4.19: Non rotated set of 16 Brodatz textures. In left-right-top-bottom order: “bark”, “brick”, “bubbles”, “grass”, “leather”, “pigskin”, “raffia”, “sand”, “straw”, “water”, “weave”, “wood” and “wool”.

Clearly, those levels containing more energy will influence more in the minimization in (4.24) and those levels with small energy will have little influence in it.

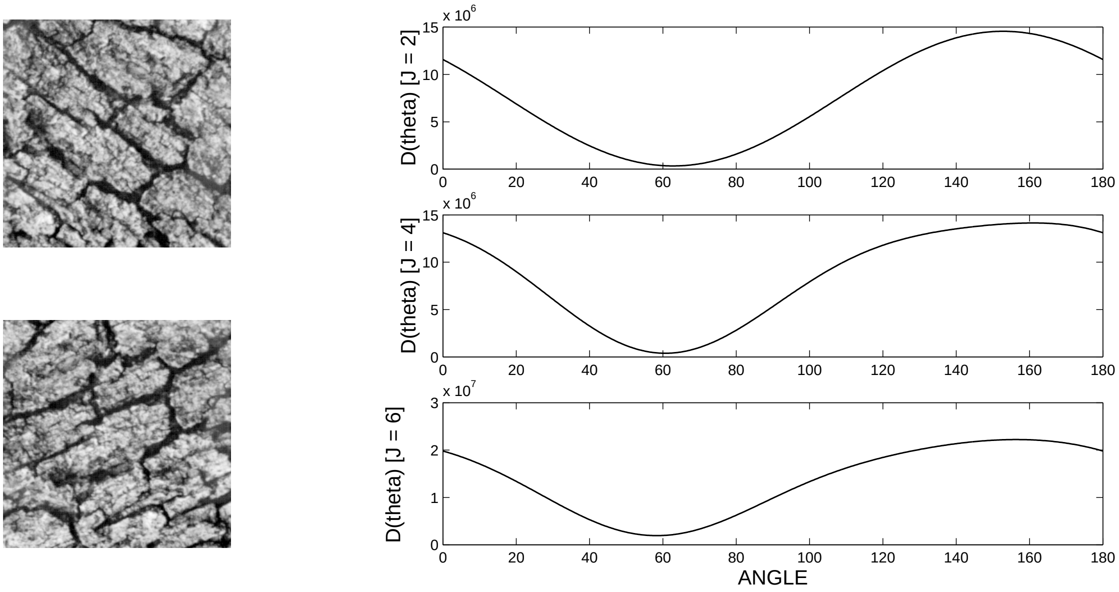
Notice that when I_1 and I_2 are two rotated versions of the same texture, the angle θ^* for which the minimum is achieved in (4.24) should be close to the relative angle between I_1 and I_2, that is, the angle you need to rotate (clockwise) I_1 in order to get I_2. Thus, one way to see the goodness of our similarity measurement (4.24) is to check whether the estimated angle θ^* is actually close to the real relative angle between 2 physically rotated versions of the same texture. Moreover, it might also be useful in some practical applications to find out approximately this relative angle.

In the next section, we show several experimental results about the estimation of relative angles between 2 texture samples of the same class and also testing our similarity measurement for the problem of rotation-invariant content based retrieval.

4.6 Experimental results: Estimation of relative angles and Rotation-Invariant retrieval

Fig. 4.19 shows the non-rotated set of textures that we have considered in our work. The complete set consists of thirteen 512×512 Brodatz texture images [85] that were rotated to different angles *before* being digitized. Fig. 4.19 shows only the non-rotated textures which can be considered without loss of generality to be oriented at 0 degrees. This basic textured images have been rotated to 6 other angles, namely, 30, 60, 90, 120, 150 and 200, obtaining 6 additional textures of the same type. Since we are interested in the range of angles from 0 to 180 and the average energy function $E^l(\phi)$ is π-periodic, a rotation of 200 is seen[9] as a rotation of 200 mod 180 = 20.

We have first tested our similarity measurement by estimating, for each class, the relative angles between the texture (512×512) images oriented at 30, 60, 90, 120, 150 and 20, and the non-rotated texture oriented at 0 degrees.

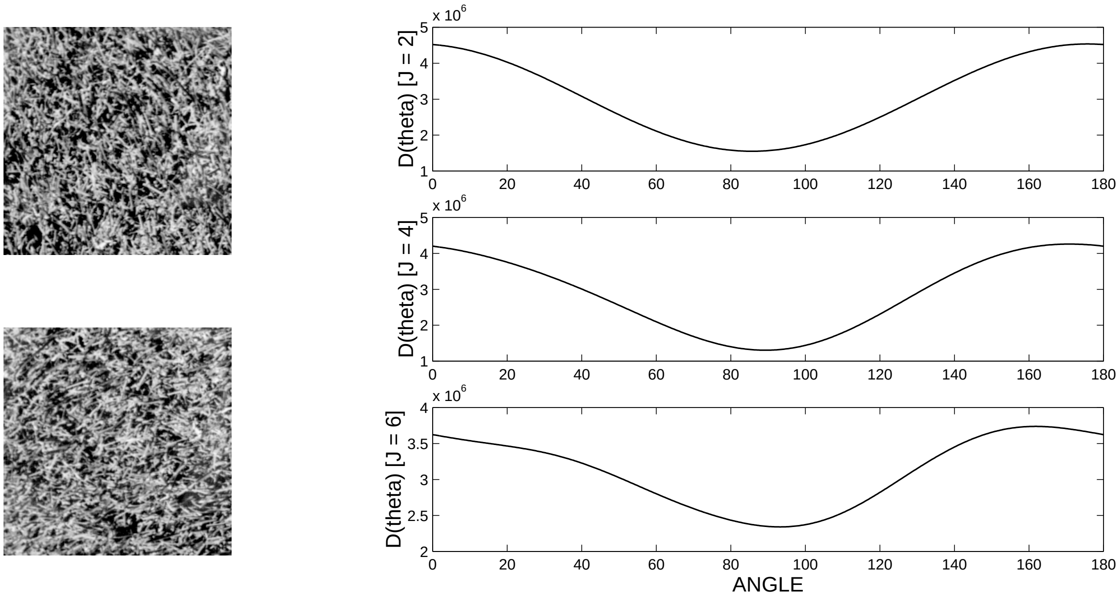


(a) (b)

Figure 4.20: (a) "bark" at 60 and 120 degrees; (b) $D(\theta)$ for $J = 2, 4, 6$

Figs. 4.20-4.23 illustrate this by showing the function $D(\theta) = \sum_{l=1}^{L} \|\boldsymbol{C}_{I_1}^l - \boldsymbol{R}(-\theta)\boldsymbol{C}_{I_2}^l \boldsymbol{R}^T(-\theta)\|_F$, for the cases where I_1 and I_2 are rotated versions of "bark", "grass", "raffia" and "weave" respectively. It can be seen that even in cases where a texture is homogeneous but does not have a predominant orientation (isotropy)

[9] Although this is a limitation of using the energy as a feature, it is not a problem for our application to not be able to distinguish between ϕ and $180 + \phi$



(a) (b)

Figure 4.21: (a) "grass" at 30 and 120 degrees; (b) $D(\theta)$ for $J = 2, 4, 6$

such as "grass" (Fig. 4.21) and "raffia" (Fig. 4.22), the estimated angle θ^* is quite close to the real relative angle between I_1 and I_2. In the same way, for textures which are not homogeneous, such as "bark" (Fig. 4.20) or "weave" (Fig. 4.23), the estimated angles are also satisfactory. In the particular case of "weave", the presence of 2 clear predominant orientations causes the appearance of 2 local minimum in the curve $D(\theta)$, however, the global minimum is very close to the correct relative angle.

Tables 4.5 and 4.6 show a summary of the estimated angles for all the different types of textures. It can be observed that for those textures that have more than 1 predominant orientation or are quite isotropic, such as "raffia", "weave", "sand" or "wool", $J = 2$ gives worse estimates than $J = 4$ or $J = 6$. In these cases, the

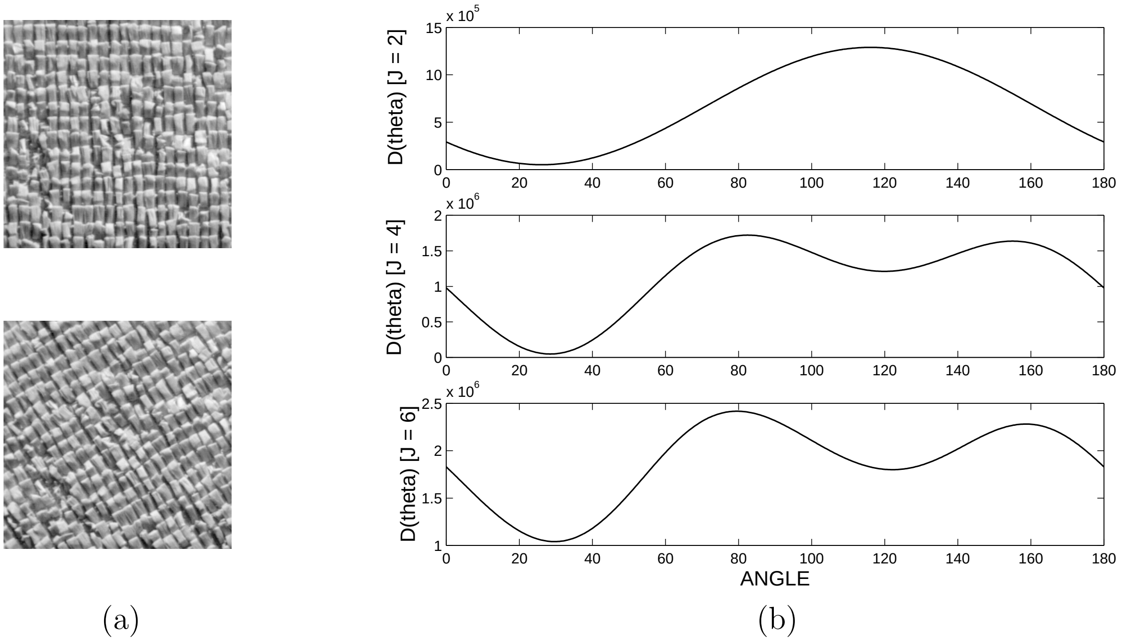
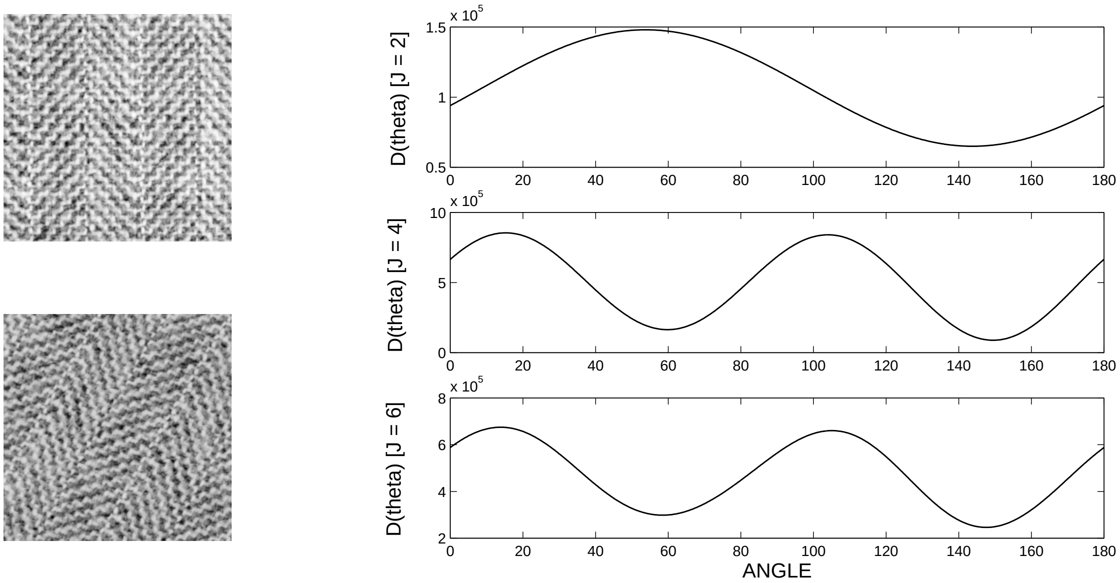


Figure 4.22: (a) "raffia" at 90 and 120 degrees; (b) $D(\theta)$ for $J = 2, 4, 6$

estimates obtained with $J = 2$ are not good in part because the resolution in angle is too small. The quality of the estimated angles for $J = 4$ and $J = 6$ are quite similar on average. In general, the estimates are reasonably good, being better for those textures having clear predominant orientations (any value of J gives similar estimates) and being worse for isotropic textures. However, there is a clear failure for the "bubbles" texture, even with $J = 6$. We believe that the main reason for this is that "bubbles" is a texture that is extremely inhomogeneous with respect to orientation, that is, it is locally oriented but the dominant orientation changes dramatically and abruptly from one part of the texture to another.

In order to test the rotation-invariance of our proposed scheme, we use 2 collections of texture subimages of size 128 × 128. The first collection, which



(a) (b)

Figure 4.23: (a) "weave" at 0 and 150 degrees; (b) $D(\theta)$ for $J = 2, 4, 6$

forms the *non-rotated* image database is obtained by partitioning each of the 13 Brodatz (512×512) textures oriented at 0 degrees (non-rotated) into 16 non-overlapping subimages (texture samples) of size 128×128. Thus, in total, we have 208 texture samples of size 128×128, 16 for each class. The second collection, which forms the *rotated* image database, is obtained by partitioning, for each of the 13 texture classes, 4 large texture images oriented at 30, 60, 90 and 120 degrees also into non-overlapping subimages of size 128×128 and taking, for each large texture image, the 4 central subimages. In this way, in the second database, for each texture class, there are also 16 textures for each class and therefore, also the same total of 208 textures.

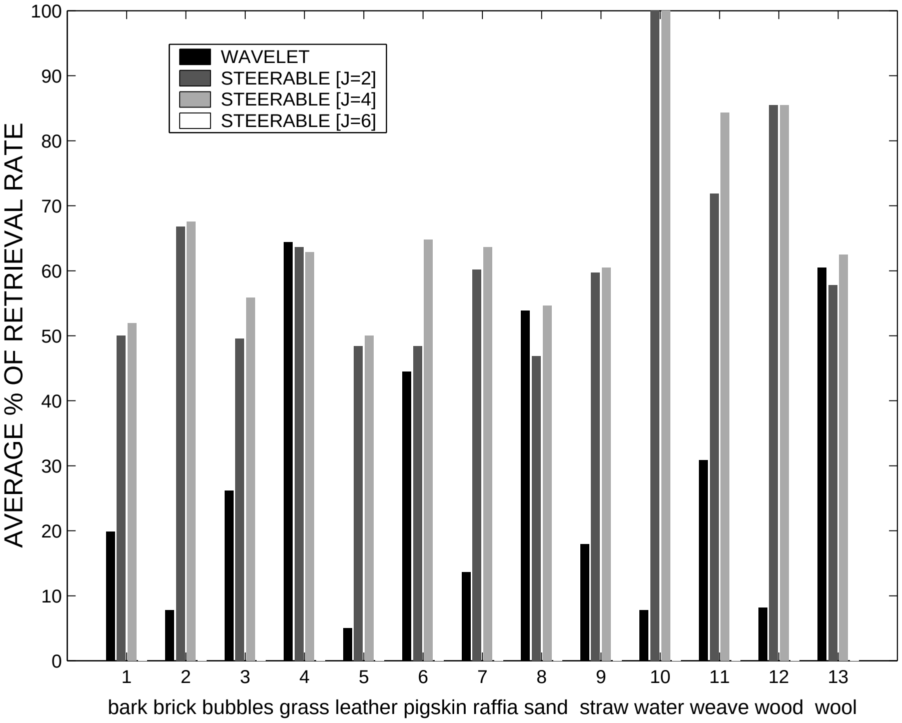


Figure 4.24: Average Percentages (%) of correct retrieval rate

In this system, a query texture sample is taken from the second database and the $M = 16$ closest textures belonging to the first *non-rotated* database are obtained using the similarity measurement given in (4.24). In the ideal case, the 16 closest textures will belong to the same class as the class of the query texture. In practice a certain number $N_c < 16$ will be obtained giving a certain percentage $100 \times N_c/16$. The performance for each class is measured in terms of the average percentage that is obtained after using as a query all the texture samples we have for a class in the second database. For comparison, we have used a three level wavelet pyramid with the 'daub3' orthogonal set and have used as features the correlation matrices obtained from the subbands at each level. As similarity measurement, we have used the total squared Frobenius distance between the two sets of three correlation matrices corresponding to the two textures that are compared.

Fig. 4.25-4.33 show examples of retrieval for several classes using a steerable pyramid with $J = 4$ (equispaced basic angles), where the texture in the top is the texture query sample from the *rotated* database and the 16 texture below are the 16 closest texture samples that have been found in the *non-rotated* database. Fig. 4.24 shows the average percentages obtained in the retrievals for each texture class in the different cases. For the different values of J, the basic angles in the experiments have been always taken to be equispaced. First, it is observed clearly that the performance obtained using the wavelet pyramid is considerably worse than that obtained with steerable pyramids, as expected because it is not possible to achieve rotation invariance with a regular wavelet transform. There are only 2 exceptions, the “sand” and “grass” textures, which are the most isotropic textures and thus, the selectivity in orientation does not provide a clear advantage.

It can be seen that although some percentages are not high, the retrievals given in Fig. 4.25-4.33 show that the texture samples that are retrieved and which belong to a different class than the query, are in many cases, perceptually similar to the target query. This is observed, for instance, in Fig. 4.27, Fig. 4.28, Fig. 4.29, Fig. 4.32 and Fig. 4.33. Finally, it should be also noted that there is a clear improvement from $J = 2$ to $J = 4$ throughout all the texture classes. On the other hand, from $J = 4$ to $J = 6$, the performance decreases for isotropic and homogeneous textures while it increases for textures that have predominant orientations and also for inhomogeneous textures such as “bark” or “bubbles”. The overall average correct retrieval rates are 27.76%, 62.23%, 66.49% and 67.85% for wavelet, $J = 2$, $J = 4$ and $J = 6$ respectively. Therefore, the best overall average correct retrieval rate is achieved by $J = 6$.

4.7 Conclusions

The basic results presented in this chapter are as follows. After a detailed review of the basic concepts, properties and construction of steerable filters using Lie theory, we first define what we call *oversteering* (angular oversampling), that is, the situation where the number of oriented subbands in the representation is larger than the minimum required to reconstruct the original image. Then, we show how this angular oversampling permits to localize most of the energy of the image in a few coefficients. Next, we show how to make use of the oversteering in order to decrease the error in the transform coefficients (increase accuracy) after these coefficients have been quantized. We describe to methods, one based on POCS theory and the other one based on calculating regions of uncertainty and its intersections. We establish several theoretical results regarding the properties of the regions of uncertainty. This second method requires to find steering functions for any set of basic angles which can be done easily using Lie Theory principles. Several experimental results are given showing the trade-off between rate and MSE as the number of orientations is increased. Next, we turn to the problem of rotation-invariant texture retrieval where we consider a simple architecture with two main tasks: feature extraction and similarity measurement. First, we describe the feature extraction which uses a steerable pyramid and calculates the correlation between all the different pairs of oriented subbands in each level in addition to the energy in each oriented subband. Then, we show that these features are actually *steerable features* in the sense that given the features corresponding for a texture oriented at a certain orientation, we show how to calculate approximately the features of the same texture but rotated to a different orientation. Finally, we show several experimental results comparing the use of a regular wavelet pyramid with a steerable pyramid using queries that are texture samples that are rotated versions of the textures that are present in the database. These experimental results show the clear superiority of using *steerable* features versus *non-steerable* wavelet features when it is necessary to have rotation-invariance.

As a final comment, we have not addressed in this chapter the complexity of performing the similarity measurement. This is being studied in our current research (see Chapter 5).

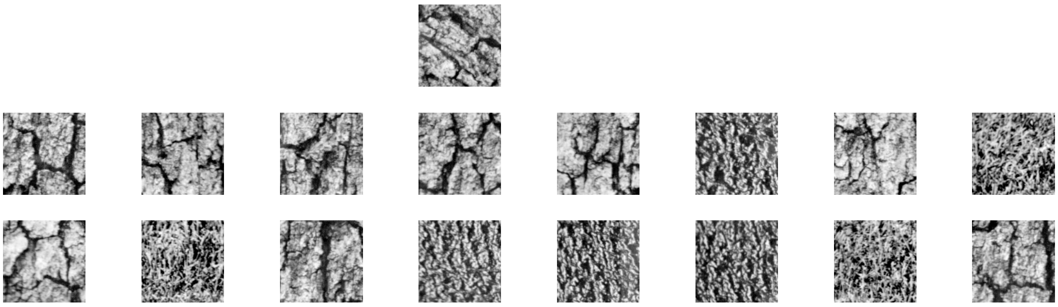


Figure 4.25: Retrieval example for “bark” at 60 degrees.

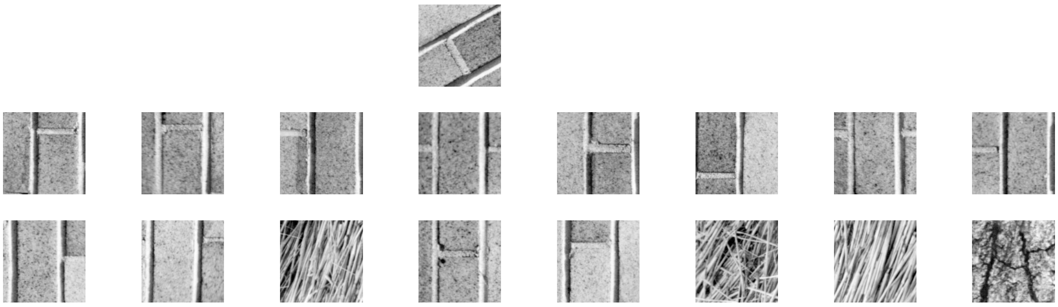


Figure 4.26: Retrieval example for “brick” at 120 degrees.

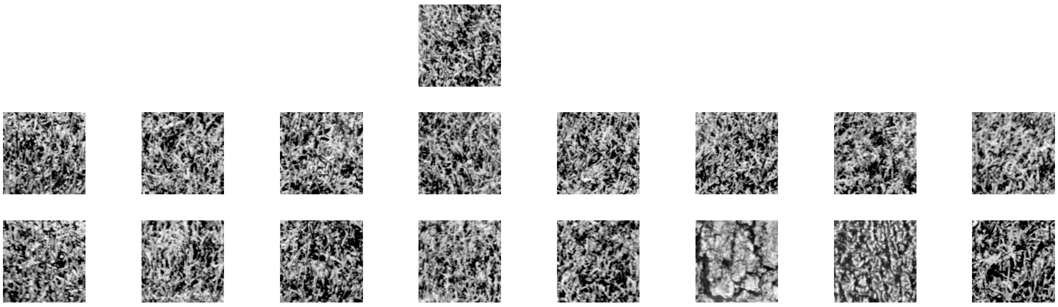


Figure 4.27: Retrieval example for “grass” at 30 degrees.

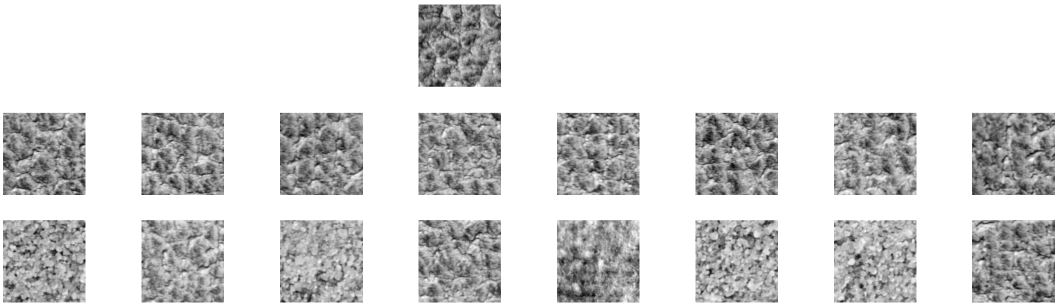


Figure 4.28: Retrieval example for “pigskin” at 90 degrees.

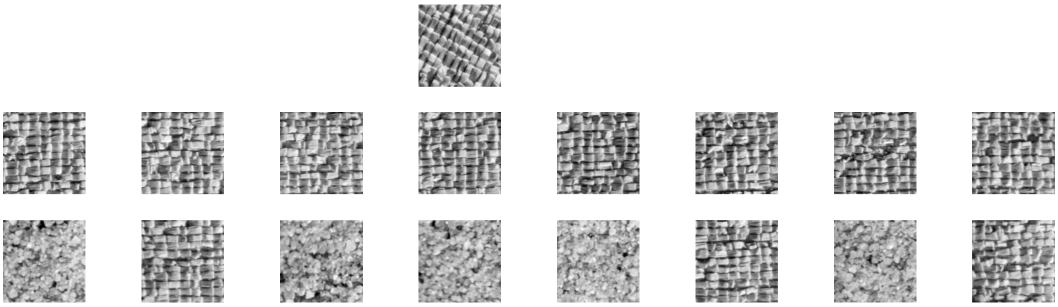


Figure 4.29: Retrieval example for “raffia” at 60 degrees.

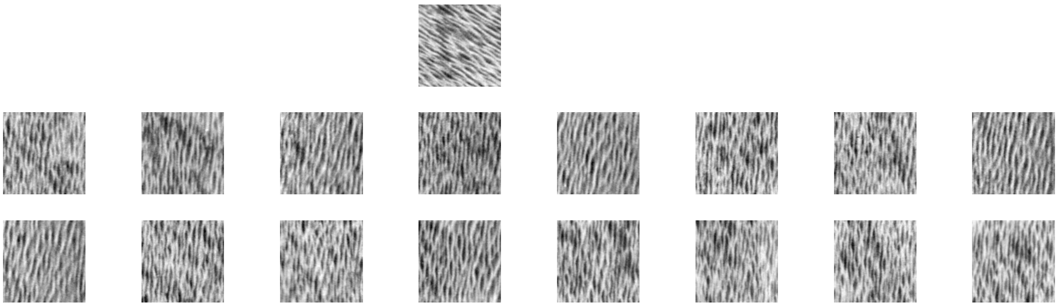


Figure 4.30: Retrieval example for “water” at 60 degrees.

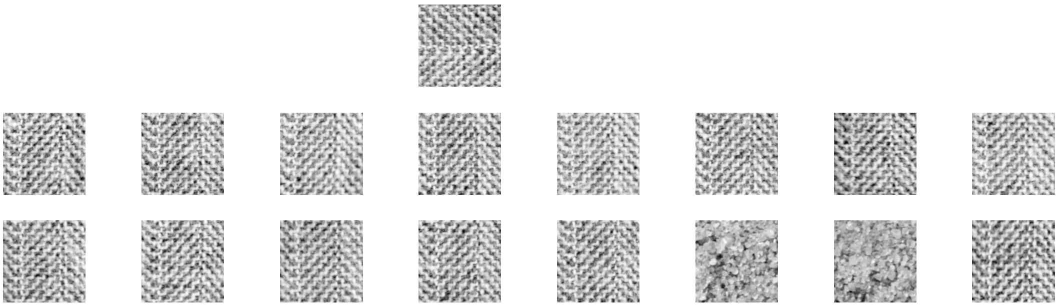


Figure 4.31: Retrieval example for “weave” at 90 degrees.

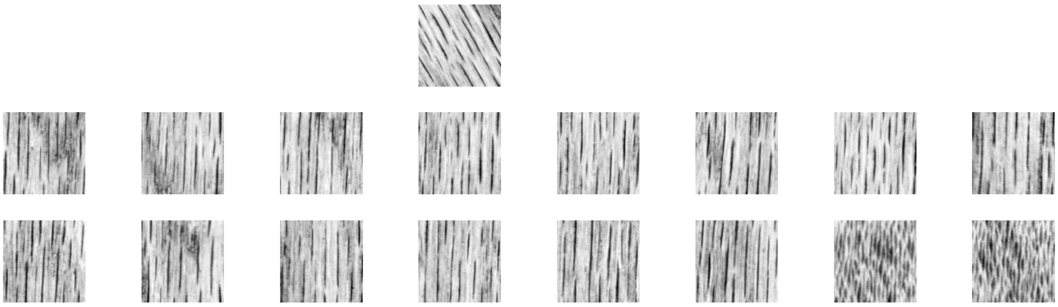


Figure 4.32: Retrieval example for “wood” at 30 degrees.

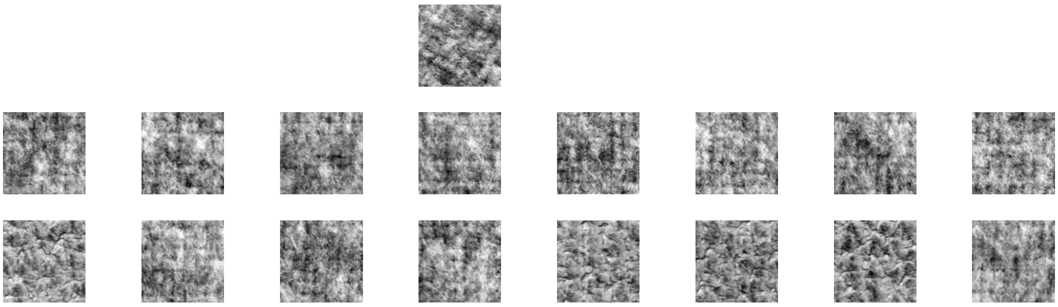


Figure 4.33: Retrieval example for “wool” at 60 degrees.

Texture Class	J	30.00	60.00	90.00	120.00	150.00	20.00
bark	2	33.17	62.74	92.82	125.48	154.54	19.80
bark	4	30.85	61.45	92.05	121.88	151.71	18.00
bark	6	28.80	60.94	92.05	119.06	151.97	16.46
brick	2	29.57	60.68	89.74	119.57	149.65	20.57
brick	4	30.08	60.42	89.74	120.08	149.40	20.05
brick	6	29.31	61.20	89.74	119.57	149.91	19.54
bubbles	2	66.60	126.00	167.14	10.02	75.34	118.28
bubbles	4	93.08	119.05	167.91	4.88	78.42	109.02
bubbles	6	93.08	106.71	169.20	179.48	80.74	98.48
grass	2	31.11	59.14	100.80	117.00	123.94	8.22
grass	4	31.62	59.65	99.51	120.60	129.60	10.54
grass	6	33.42	58.89	94.11	125.74	135.77	15.68
leather	2	30.34	58.88	85.11	118.80	149.65	22.88
leather	4	31.37	60.42	85.11	119.31	150.68	23.91
leather	6	29.31	58.11	83.57	117.00	146.57	23.40
pigskin	2	33.68	48.34	78.68	100.80	147.60	31.62
pigskin	4	32.40	52.20	81.51	103.88	151.20	28.02
pigskin	6	29.31	51.43	82.03	102.09	147.85	24.17

Table 4.5: Estimated angles for different textures where I_1 is a type of (non-rotated) texture at 0 degrees and I_2 is the same texture as in I_1 but rotated at 30, 60, 90, 120, 150 and (200 mod 180 = 20) degrees .

Texture Class	J	30.00	60.00	90.00	120.00	150.00	20.00
raffia	2	19.02	45.51	75.08	101.31	136.54	20.05
raffia	4	29.57	59.14	88.45	116.22	148.11	18.25
raffia	6	29.31	59.14	89.48	116.23	147.85	17.49
sand	2	39.85	63.77	103.62	135.77	136.28	175.62
sand	4	37.28	65.31	101.82	129.85	141.68	4.37
sand	6	32.40	60.94	98.49	126.77	139.11	6.95
straw	2	27.00	54.51	89.22	118.28	147.85	18.77
straw	4	28.02	55.28	89.74	118.80	148.62	20.05
straw	6	28.55	56.06	89.74	119.31	148.89	20.06
water	2	28.02	58.62	87.17	117.25	148.11	18.00
water	4	28.28	58.88	87.68	117.00	148.62	18.25
water	6	28.80	58.89	87.94	118.02	148.37	19.03
weave	2	43.71	53.74	83.57	110.57	143.74	7.20
weave	4	33.17	59.65	90.77	122.40	149.40	24.42
weave	6	33.94	58.63	90.51	122.91	148.63	24.42
wood	2	28.54	62.22	89.48	118.80	150.68	19.54
wood	4	28.80	61.45	89.48	118.80	149.91	18.77
wood	6	28.80	61.46	89.49	118.54	149.40	18.51
wool	2	40.11	65.31	116.22	115.97	152.74	27.77
wool	4	31.11	64.02	93.34	115.20	153.00	20.05
wool	6	28.80	64.28	91.80	110.31	154.28	16.20

Table 4.6: Estimated angles for different textures where I_1 is a type of (non-rotated) texture at 0 degrees and I_2 is the same texture as in I_1 but rotated at 30, 60, 90, 120, 150 and (200 mod 180 = 20) degrees.

Chapter 5

Current and Future work

Our future work comprises several topics which are related in some way to one of these main areas: a) Quantized overcomplete expansions and its applications in quantization, signal processing and communications, b) Lattice theory and its applications in communications, quantization and code design for network problems and c) shiftable or steerable filter banks and its several applications to signal processing.

On the one hand, some of the proposed directions try to continue the work explained in the previous chapter. On the other hand, several other topics are proposed which are not a direct continuation of the work contained in this thesis but are related to some of the different topics described in this thesis.

Next, we explain each of these topics. Some specific topics are actually object of our current research and thus, are explained in much more detail than others.

5.1 Amalgamation and Dithering in Periodic Quantizers

There are two possible ideas to try to improve the performance of these multiple description quantizers:

1. The idea of merging (amelgamating) cells should be studied carefully. As it was mentioned in Chapter 3, for the case shown in Fig. 3.2, the diagonally and vertically shaded triangles could be merged to give a tesselation made up of regular hexagons and equilateral triangles with the same edge length as the hexagons. The new tesselation will have a larger *absolute* mean squared error but a smaller (dimensionless) *normalized* mean squared error G, which means that its rate-distortion performance is better. In other words, if we merge 2 bad (not having a good shape) cells into a good cell, we are increasing gracefully the absolute mean square error but we are decreasing the resulting rate necessary to encode this tesselation, which intuitively should give rise to a better normalized mean squared error G. We have not investigated this possibility.

2. Another idea is the use of dithering, which in this context translates to allowing arbitrary shifts in each of the lattices $\Lambda^1, \cdots, \Lambda^r$ (corresponding to the individual quantizers) before performing the actual quantization.

> It should interesting to study the design of quantizers by taking a simple initial lattice Λ^1 and combining it with several other lattices which are now both rotations *and translations* of Λ^1, instead of only rotations as we have considered in Chapter 2 and Chapter 3.

Notice that even though we have not obtained favorable results in the 4 examples we have tried, it is still not clear whether it is possible or not to design good quantizers by using periodic quantizers based on lattice intersections. It should be taken into account that (non-lattice) tesselations which are better than the best so-far known lattice tesselations have been already found in certain dimensions [62]. On the other hand, the research of these quantizers in the context of applications such as A/D conversion (quantized overcomplete expansions) may still be interesting.

5.2 Connections between Lattice Intersections and Lattice Sampling Conversion

Given a multidimensional continuous signal $\boldsymbol{x}_c(t_1, \cdots, t_N) = [x(t_1), \cdots, x(t_N)]$ that has been sampled in a lattice Λ^1 giving vectors $\boldsymbol{x}_s(i_1, \cdots, i_N) = [x(i_1), \cdots, x(i_N)]$, it is often necessary to resample the same signal in a different lattice Λ^2

without having to reconstruct the original continuous signal $\boldsymbol{x}_c$. The following steps are taken in order to do this [46]:

1. Upsampling of $\boldsymbol{x}_s$ from Λ^1 to $\Lambda^1 + \Lambda^2$ obtaining[1] $\boldsymbol{x}_s^u$.

2. Low pass filtering of $\boldsymbol{x}_s^u$ with a multidimensional filter $L(\boldsymbol{\omega})$ whose pass band region in the multidimensional Fourier domain is the Voronoi region around $\mathbf{0}$ of the lattice $(\Lambda^1 + \Lambda^2)^*$, hence, the periodicity in the spectrum domain of the filtered signal is determined by the lattice $(\Lambda^1 + \Lambda^2)^*$.

3. Finally, the output of the low pass filtering is downsampled to the lattice Λ^2.

The overall conversion rate is therefore given by $det(M_{\Lambda^2})/det(M_{\Lambda^1})$. A very important issue is that in order to avoid this system being shift-variant, which is very undesirable in any signal processing application, the lattices Λ^1 and Λ^2 must have a *non-empty intersection.* In this latter case, the whole system becomes shift-invariant with respect to the intersection lattice $\Lambda^1 \cap \Lambda^2$. Similar conclusions are obtained it the upsampling is performed with respect to a sum of more than 2 lattices, which allow at the other end, to perform downsampling in more than one lattice.

In all the constructions given in Chapter 3, we have several lattices $\Lambda^1, \cdots, \Lambda^r$ which are rotated versions of each other whose intersection gives a lattice Λ.

[1] As explained in Chapter 2, $\Lambda^1 + \Lambda^2$ may not be a lattice but just a certain set of points.

Thus, our decompositions provide all the different (up to scaling) possible lattice conversion structures while keeping the shift-invariance always with respect to the same (intersection) lattice Λ. It would be interesting to study whether this fact is useful in a) obtaining new interpolation methods for multidimensional signals, b) distributed sampling applications where the same continuous multidimensional signal is sampled in different lattices and all or some of these sampled versions have to be combined appropiately to estimate the original signal with more resolution in certains parts of the signal domain than in others.

5.3 Optimal power shaping for the Costa problem ("Writing on Dirty Paper")[2]

5.3.1 Basic description of the Costa problem

Let $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{S}$ and $\boldsymbol{Z}$ be random N-dimensional vectors. The Costa ("Writing on Dirty Paper") problem models the channel as:

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{S} + \boldsymbol{Z} \tag{5.1}$$

[2] This is current work being carried out in collaboration with Suhas Diggavi from AT&T Shannon Laboratory. Part of this work is to be published in[12].

where $\boldsymbol{X}$ is the transmit vector signal, $\boldsymbol{Y}$ is the receive vector signal, $\boldsymbol{S}$ is the interference vector signal, and $\boldsymbol{Z}$ is a Gaussian random vector. The interference signal $\boldsymbol{S}$ is non-causally known at the transmitter but it is not known at the receiver, and the noise $\boldsymbol{Z}$ is independent of $\boldsymbol{X}$ and $\boldsymbol{S}$ and it is not known neither the transmitter nor the receiver. The transmit signal must satisfy a power constraint, that is, $\frac{1}{N}E[\|\boldsymbol{X}\|^2] \leq P$.

When the interference and noise are joint i.i.d. Gaussian variables, Costa [36] showed using random binning methods that if the entire set of samples of the interfering signal is known to the transmitter in a *non-causal* way (key condition), the capacity of this channel is given by:

$$C = \frac{1}{2}\log\left(1 + \frac{P}{\sigma_Z^2}\right) \tag{5.2}$$

where σ_Z^2 is the variance of the gaussian random variable Z. Thus, the effect of the interference S is completely cancelled, that is, as if there were not interference or the interference were known also at the receiver. This result has been extended in different ways. For the vector case [29, 134] as in (5.1) where $\boldsymbol{S}$ and $\boldsymbol{Z}$ are i.i.d. Gaussian vector signals, a similar result holds and the capacity of the channel is given by:

$$C = \frac{1}{2}\log\frac{det(\boldsymbol{K_X} + \boldsymbol{K_Z})}{det(\boldsymbol{K_Z})} \tag{5.3}$$

where $\boldsymbol{K}_{\boldsymbol{X}}$ and $\boldsymbol{K}_{\boldsymbol{Z}}$ are the covariance matrices of $\boldsymbol{X}$ and $\boldsymbol{Z}$ respectively. This is the mutual information for the vector Gaussian channel without any interfering signal $\boldsymbol{S}$. Costa's result has been also generalized to cases where the interfering signal is an ergodic process [29] with an arbitrary distribution. Zamir, Shamai and Erez [137] have generalized the result to the case when the interfering signal path is an arbitrary sequence and a common source of randomness (dithering) is used which is available to both the transmitter and the receiver.

There are several applications which are directly connected to the Costa problem. A very important problem is that of non-degraded vector (multi-antenna) wireless broadcast channels when the transmitter is equipped with multiple antennas [20, 132]. Another example appears in digital subscriber lines (DSL) where there is electromagnetic coupling between different lines, which can also be modeled as a vector broadcast channel [133]. Other examples of applications include digital watermarking, multimedia information-hiding, steganography [24] and intersymbol interference (ISI) cancellation [137].

We first provide a review of some related work on code constructions for the Costa problem and explain briefly the novelty in our work. Then, we explain the baseline construction in order to illustrate the basic idea and in Section 5.3.4, we analyze its limitations in terms of power shaping, motivating the use of a trellis to increase dimension with low complexity and explain with more detail the difference with our work. Next, in Section 5.3.5, we review the trellis precoding idea [49] where the power shaping is performed through constellation points and emphasize with more detail the difference with our work. In Section 5.3.6, we introduce our new approach of performing the power shaping through a sequence of quantizers that is chosen through a trellis structure and finally, in Section 5.3.7, we show how to perform shaping in a joint manner, more specifically, we show how to combine trellis precoding and our approach by using the tensor product of two trellises. We provide a detailed example.

5.3.2 Code constructions for the Costa problem

In all practical constructions it is always assumed, for complexity reasons, that the encoder has a non-causal knowledge about the interference signal, but the non-causality is always finite, that is, the encoder only knows the current realization of the random interference vector $\boldsymbol{S}$ of a certain finite dimension. All the constructions, inspired by the random binning proof of Costa's result, use two nested codes, where the fine code plays the role of a channel code and the coarse code plays the role of a source code or quantizer.

Erez, Zamir and Shamai [137] have proposed a lattice based scheme with 2 nested lattices which makes an explicit use of a source of randomness and dithered quantization, both at the encoder and the decoder side. The adaptativity of the system to the allowed power P and the noise power σ_Z^2 is obtained through the use of an (asymptotically optimal) estimation coefficient $\alpha = P/(P+\sigma_Z^2)$ (where $\boldsymbol{K_Z} = \sigma_Z^2\boldsymbol{I}$) and where the interference signal $\boldsymbol{S}$ is scaled with this coefficient. However, they do not present any examples of concrete constructions and only show the existence of lattices through a random construction based on a Loeliger's construction [77]. Chou, Pradhan and Ramchandran [25] have proposed turbo coded trellis-based constructions for this problem. The interference $\boldsymbol{S}$ is also assumed to be a white and Gaussian vector and the same scaling is performed to adapt to different levels of noise and transmitting power.

Eyuboglu and Forney [49] proposed a method called Trellis precoding for intersymbol interference (ISI) channels, which under the assumption of channel state information at the encoder, reduces to the Costa problem as pointed out in [137]. This work, which makes use of nested cubic (fine and coarse) lattices, introduces for the first time the idea of performing a (low complexity) shaping in the power using a trellis structure similar to the trellis shaping idea introduced by Forney [50].

We are interested in giving concrete lattice constructions and analyze the practical performance of theses sytems such as the supported bit rates and the probability P_e of decoding error for a certain finite dimension, noise power σ_Z^2, rate R and allowed power P at the transmitter and evaluate this performance in terms of the geometric parameters of concrete lattices or by simulation. Our focus is also on deterministic quantization rather than dithered quantization and the adaptability to the the different levels of power and noise is obtained by scaling properly the lattices while keeping them nested. We do not assume any statistical model for the interference $\boldsymbol{S}$ while the noise is assumed to be a white Gaussian vector. Although it is possible to use channel trellis codes on top of the lattice constructions in order to increase the coding gain of these systems against the noise, we do not consider this issue here and the main concern in this work is to perform a good *shaping* to save transmitting power as in [49].

The main difference with the scheme introduced by Eyuboglu and Forney is that there the shaping is obtained through the fine lattice by expanding the basic constellation, hence, reducing the rate. We propose the idea of performing shaping through the coarse lattice which do not result in a constellation expansion. More specifically, we introduce the new idea of transmitting information bits through a constrained sequence of lattice coset quantizers and explore the use of very simple coarse lattices in order to have as low complexity as possible. On the other hand, our method can be actually combined with the shaping method in [49], so that the shaping is jointly performed by both the channel (fine) code and by the source (coarse) code.

5.3.3 Basic (Baseline) Construction

Let Λ^1 and Λ^2 be two N-dimensional lattices in $\mathbb{R}^N$ such that $\Lambda^2 \subset \Lambda^1$, that is, Λ^2 is a sublattice of Λ^1. If $\boldsymbol{M}_{\Lambda^1}$ and $\boldsymbol{M}_{\Lambda^2}$ are the respective generator matrices, then the index $|\Lambda^1/\Lambda^2|$ of Λ^2 in Λ^1 is given by $det(\boldsymbol{M}_{\Lambda^2})/det(\boldsymbol{M}_{\Lambda^1})$ and is equal to the number of cosets of Λ^2 inside Λ^1. Let $|\Lambda^1/\Lambda^2| = L$. Then, we can identify L coset (canonical) representatives $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_L\}$ which are inside the fundamental cell $C_{\Lambda^2}(\boldsymbol{0})$ of the sublattice Λ^2 that has the $\boldsymbol{0}$ vector as its center. This set of points is given by $\Lambda^1 \cap C_{\Lambda^2}(\boldsymbol{0})$. In cases where the sublattice Λ^2 is not clean, that is, there are border points falling on the envelope of $C_{\Lambda^2}(\boldsymbol{0})$, the mapping of these points is done in a systematic way ensuring that $(\boldsymbol{v}_1 + \Lambda^2) \cup \cdots \cup (\boldsymbol{v}_L + \Lambda^2) = \Lambda^1$. Each coset $\boldsymbol{v}_i + \Lambda^2$ is to be associated uniquely with a different message $\boldsymbol{m}_i$, that is, all points in the coset $\boldsymbol{v}_i + \Lambda^2$ are equivalent and are associated to the message $\boldsymbol{m}_i$. Thus, the transmitting rate R per sample is simply given by $R = \frac{1}{N}\log_2(|\Lambda^1/\Lambda^2|)$.

For purposes of normalization and for reasons that will become clear later, let us assume that $\Lambda^1 = \frac{\alpha}{k}\Lambda^1_o$ where $\alpha \in \mathbb{R}_+$, $k \in \mathbb{Z}_+$ and $Vol(\Lambda^1_o) = det(\boldsymbol{M}_{\Lambda^1_o}) = 1$, and also $\Lambda^2 = \alpha\Lambda^2_o$, where Λ^2_o is the corresponding sublattice of the normalized lattice Λ^1_o satisfying that $Vol(\Lambda^2_o) = det(\boldsymbol{M}_{\Lambda^2_o}) = L$. Notice that under these conditions, for any $k \in \mathbb{Z}_+$, $\Lambda^2 \subset \Lambda^1$. Let Q_Λ be the lattice vector quantizer defined by the lattice Λ. The baseline system works as follows:

Encoding Procedure

1. Select coset representative (symbol) $\boldsymbol{v}_i$ associated with the message $\boldsymbol{m}_i$ which is intended to be transmitted.

2. Given the current known interference vector $\boldsymbol{s}$, the vector signal that is transmitted is given by:

$$\boldsymbol{x} = (\boldsymbol{v}_i - \boldsymbol{s}) - Q_{\Lambda^2}(\boldsymbol{v}_i - \boldsymbol{s}) = (\boldsymbol{v}_i - \boldsymbol{s}) \quad mod \quad \Lambda^2 \tag{5.4}$$

Hence, what is transmitted is the quantization error resulting from quantizing the signal $(\boldsymbol{v}_i - \boldsymbol{s})$ with respect to the coarse sublattice Λ^2. The channel adds the interference $\boldsymbol{s}$ and the noise $\boldsymbol{z}$ vectors to the transmitted signal $\boldsymbol{x}$, so the decoder receives:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{s} + \boldsymbol{z} = \boldsymbol{v}_i - Q_{\Lambda^2}(\boldsymbol{v}_i - \boldsymbol{s}) + \boldsymbol{z} \tag{5.5}$$

Notice that in the absence of the noise vector $\boldsymbol{z}$, the vector $\boldsymbol{y}$ is just the coset representative $\boldsymbol{v}_i$ shifted by a point in the sublattice Λ^2, resulting in a point that falls in the same coset $\boldsymbol{v}_i + \Lambda^2$. Since each message $\boldsymbol{m}_i$ is associated with a unique coset, the effect of the interference has been actually cancelled out.

Decoding Procedure

1. Quantize the received vector $\boldsymbol{y}$ with respect to the fine lattice Λ^1, obtaining $\boldsymbol{b} = Q_{\Lambda^1}(\boldsymbol{y})$.

2. The reconstructed vector $\hat{\boldsymbol{v}}$ is finally obtained performing a modulo-Λ^2 operation on the vector $\boldsymbol{b}$:

$$\hat{\boldsymbol{v}} = \boldsymbol{b} - Q_{\Lambda^2}(\boldsymbol{b}) = \boldsymbol{b} \quad mod \quad \Lambda^2 \tag{5.6}$$

In this construction, the fine lattice Λ^1, for the given dimension, should be a good (channel code) lattice in the sense that its coding gain should be as high as possible. On the other hand, the power of the transmitted signal $\boldsymbol{x}$ is controlled by the coarse sublattice Λ^2 because it can be seen from the encoding procedure that $\boldsymbol{x} \in C_{\Lambda^2}(\mathbf{0})$ and thus, the power spent by the transmitter is equal to the power of the quantization noise that is generated from quantizing $(\boldsymbol{v}_i - \boldsymbol{s})$ with respect to Λ^2.

It is easy to show that as the number L of independent signal points $\{\boldsymbol{v}_i\} \in C_{\Lambda^2}(\mathbf{0})$ increases and assuming that these signal points are used equiprobably, the distribution of the transmitted signal $\boldsymbol{x}$ becomes very well approximated by an i .i.d. uniform N-dimensional distribution over the cell $C_{\Lambda^2}(\mathbf{0})$ [80]. Under this assumption, the power P per dimension will be given by:

$$P = \frac{1}{N}\frac{1}{Vol(\Lambda^2)}\int_{C_{\Lambda^2}(\mathbf{0})} \|\boldsymbol{r}\|^2 d\boldsymbol{r} = G(\Lambda^2)(Vol(\Lambda^2))^{2/N} = \alpha^2 L^{2/N} G(\Lambda_o^2) \tag{5.7}$$

where $G(\Lambda^2)$ is the normalized second moment of the lattice quantizer Q_{Λ^2} and which is defined as:

$$G(\Lambda) = \frac{\int_{C_\Lambda(\mathbf{0})} \|\boldsymbol{r}\|^2 d\boldsymbol{r}}{N(Vol(\Lambda))^{1+2/N}} \tag{5.8}$$

On the other hand, the supported rates per sample for this system are given by:

$$R = \frac{1}{N}\log_2\left(\frac{Vol(\Lambda^2)}{Vol(\Lambda^1)}\right) = \frac{1}{N}\log_2\left(\frac{det(M_{\Lambda^2})}{det(M_{\Lambda^1})}\right) = \frac{1}{N}\log_2(k^N L) \tag{5.9}$$

where L can take on all the possible positive index values given by all the possible sublattices Λ_o^2 which are nested in the normalized lattice Λ_o^1.

The probability P_e of decoding error is the same as for the vector Gaussian channel where the lattice Λ^1 is used for signaling, which can be approximated using the classical union bound by:

$$P_e \simeq \tau(\Lambda_o^1) Q\left(\sqrt{\frac{\alpha^2 d_{min}^2(\Lambda_o^1)}{4k^2\sigma_Z^2}}\right) \tag{5.10}$$

where $\tau(\Lambda_o^1)$ and $d_{min}^2(\Lambda_o^1)$ denote the kissing number and the squared minimum distance of the normalized lattice Λ_o^1.

Suppose that we are given a power constraint P. There are several combinations of values for α and the index $|\Lambda_o^1/\Lambda_o^2|$ which can be used to obtain the same power. The set of pairs $\{L, \alpha\}$ with $L \in \mathbb{Z}$ being a valid index and which give rise to the same power are given by $\left\{L, \sqrt{\frac{P}{L^{2/N}G(\Lambda_o^2)}}\right\}$. Thus, P_e results in:

$$P_e \simeq \tau(\Lambda_o^1) Q\left(\sqrt{\frac{1}{k^2 L^{2/N}} \frac{d_{min}^2(\Lambda_o^1)}{G(\Lambda_o^2)} \frac{P}{4\sigma_Z^2}}\right) = \tau(\Lambda_o^1) Q\left(\sqrt{\frac{1}{2^{2R}} \frac{d_{min}^2(\Lambda_o^1)}{G(\Lambda_o^2)} \frac{SNR}{4}}\right) \tag{5.11}$$

where it can be seen that since Λ_o^1 is normalized so that $Vol(\Lambda_o^1) = 1$, the term $d_{min}^2(\Lambda_o^1)$ is equal to the coding gain of the lattice Λ_o^1 and on the other hand, the term $1/G(\Lambda_o^2)$ is proportional to the shaping gain of the lattice Λ_o^2. Thus, the lattice Λ_o^2 is acting as a shaping lattice and hence determining the power while Λ_o^1 is controlling the performance against the noise.

Therefore, for a given constrained power P, if the power of the noise increases (decreases) and thus the SNR decreases (increases), in order to keep the same value for P_e, the rate R should be reduced (increased) by increasing (decreasing) $d_{min}^2(\Lambda^1)$ while keeping the nesting between the two lattices and without increasing the generated power. The value of $d_{min}^2(\Lambda^1)$ (and the rate) is controlled by two parameters, the index $|\Lambda_o^1/\Lambda_o^2| = L$ and the positive integer k which keeps the nesting property:

$$d_{min}^2(\Lambda^1) = \frac{1}{k^2 L^{2/N}} \frac{P}{G(\Lambda_o^2)} d_{min}^2(\Lambda_o^1) \tag{5.12}$$

If the power P is to be kept constant, a decrease (increase) in L has to be compensated by a corresponding increase (decrease) in the scale factor α. Notice from (5.12) that by increasing (decreasing) the product $k^2L^{2/N}$, $d^2_{min}(\Lambda^1)$ is decreased (increased). For the particular case where all the basic lattices Λ^1_o and Λ^2_o are restricted to be geometrically similar, the values of $d^2_{min}(\Lambda^1_o)$ and $G(\Lambda^2_o)$ will not change. In this case, for a target probability P_e of decoding error, one has to find numerically, for each SNR, the best pair of integers (k, L) (L restricted to be a valid index value) which better approximate P_e. Those optimal values will determine using (5.9) the maximum possible rate R per sample that can be used for that SNR. Once the optimal integers (k, L) have been found, in order to use them in practice, they can be stored in a look-up table and there is no need to search for them every time. For instance, if $\Lambda^1_o = A_2$, then, the possible values for L are given by $L = a^2 - ab + b^2$, $a, b \in \mathbb{Z}$ [30]. Substituting this value of L in (5.9), one can obtain easily a curve $R_{max}(SNR)$. Similar curves can be obtained for the important cases of $\Lambda^1_o = D_4, E_8$ using the respective allowed values for L in each case [30].

5.3.4 Shaping through a trellis structure: Motivation and proposed approach

There is a practical limitation for the baseline system in terms of the dimension N that can be used because if Λ^1 and Λ^2 are chosen to be good lattices for each dimension, the complexity of the quantizers Q_{Λ^1} and Q_{Λ^2} increases very importantly with the dimension N. Thus, it is not possible to efficiently approximate well the ultimate shaping gain of $\pi e/6$ (1.53 dB) which is achieved by a uniform distribution over an N-dimensional sphere in the limit as $N \to \infty$ whose projection in any 2 dimensions gives a perfect Gaussian distribution (optimal distribution for the Gaussian channel). This happens when $G(\Lambda_o^2) \to \frac{1}{2\pi e}$, which is the normalized second moment of an N-dimensional sphere as $N \to \infty$.

The goal of shaping is therefore to achieve a non-uniform, Gaussian-like distribution, so as to reduce the average transmitting signal power while keeping the same rate. This power reduction is the shaping gain. Using practical values of dimension, the resulting shaping gain that can be achieved with the baseline system is actually limited to a few tenths of a dB. On the other hand, in order to achieve capacity, it is necessary to generate a gaussian distribution of power.

Eyuboglu and Forney [49] proposed to use a method called trellis precoding for ISI channels where the dimension of the lattices (all of them cubic) is increased through a convolutional shaping code.

In our work, we propose a different way of performing shaping which can be used either independently of the method in [49] or in conjunction with it. For purposes of understanding, we first describe the classical trellis precoding idea. Then, we propose a new approach which uses the concept of transmission of information through a sequence of quantizers which are obtained by shifting a basic coarse lattice quantizer. We also show later how this approach can be combined and complemented with the trellis precoding method which uses a sequence of signal points of the fine lattice instead of quantizers. In this last case, as will be shown, a sequence of quantizers and signal points is jointly optimized in order to achieve the lowest possible average power and the optimization is performed using the tensor product of 2 trellises, one working on the signal points (fine lattice) and the other one working on cosets of the coarse lattice.

Notice that in order to have a low complexity at the encoder, taking into account the encoder procedure described in 5.3.3 where quantization is performed with respect to the coarse lattice, it is interesting to use coarse lattices which are as simple as possible, ideally, cubic lattices.

5.3.5 Trellis precoding idea

Let $\Lambda^1/\Lambda^2/\Lambda^3$ be a lattice partition chain where $|\Lambda^1/\Lambda^2| = 2^{n_u}$ and $|\Lambda^2/\Lambda^3| = 2^{n_p}$. Then, the set of $2^{n_u+n_p}$ lattice points belonging to Λ^1 which are inside the Voronoi cell $C_{\Lambda^3}(\mathbf{0})$ can be partitioned into 2^{n_u} non-overlapping sets (there are several ways to choose these sets, that is, there is no one unique partition), each having 2^{n_p} points of Λ^1. The shaping operation introduced by Eyuboglu and Forney consists of associating more than one signal vector with each binary message $\boldsymbol{m}$. This association is given implicitly by the lattice partition $\Lambda^1/\Lambda^2/\Lambda^3$ so that each particular message $\boldsymbol{m}$ is associated with a non-overlapping set $\boldsymbol{A_m}$ of 2^{n_p} points in Λ^1.

Let $\{\boldsymbol{m}_{i_1}, \boldsymbol{m}_{i_2}, \cdots, \boldsymbol{m}_{i_J}\}$ and $\{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_J\}$ denote the sequence of J binary messages to be transmitted and the corresponding sequence of interference vectors. The goal is to minimize the average power P per dimension given by:

$$P = \frac{1}{NJ}\sum_{j=1}^{m} \|\boldsymbol{x}_{i_j}\|^2 = \frac{1}{NJ}\sum_{j=1}^{J} \|(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j) - Q_{\Lambda^3}(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j)\|^2 \tag{5.13}$$

where $\boldsymbol{x}_{i_j}$ is the transmitting vector associated with the signal vector $\boldsymbol{v}_{i_j}$. The most immediate way to perform shaping would be to choose, for each message $\boldsymbol{m}_{i_j}$ in the sequence, the signal vector $\boldsymbol{v}_{i_j}$ in the set $\boldsymbol{A}_{\boldsymbol{m}_{i_j}}$ such that the individual term $\|(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j) - Q_{\Lambda^3}(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j)\|^2$ is minimized independently. However, this decreases the tranmitting rate from (n_u+n_p) bits per 2 dimensions to n_u bits per 2 dimensions. Thus, this results effectively in a constellation expansion factor CER of 2^{n_p}. In the trellis precoding method introduced by Eyuboglu and Forney for the ISI channel, the idea is to reduce the freedom by constraining the selection of the signal vector points through a trellis in a sequence. In this way, the resulting reduction in rate is smaller. The trellis structure is given by the trellis diagram of a shaping linear convolutional code C_p with rate k_p/n_p. In this trellis diagram, there are 2^{k_p} branches emanating from each state and all the possible paths of length J will determine all the possible choices of signal vectors which are valid for the whole sequence of J binary messages.

Notice that each of the 2^{n_p} signal points contained in a set $\boldsymbol{A_m}$ can be labeled uniquely by a binary n_p-tuple $(b_1, \cdots, b_{n_p})$. Let $\boldsymbol{G}_p(D)$ be the $k_p \times n_p$ generator matrix of C_p. Then, we can find a (non-unique) syndrome-former $n_p \times (n_p - k_p)$ matrix $\boldsymbol{H}_p^T(D)$ such that $\boldsymbol{G}_p\boldsymbol{H}_p^T = \boldsymbol{0}$. The important observation is that if $\boldsymbol{y}(D) == (y_1(D), \cdots, y_{n_p}(D)) = \{(y_1^1, y_1^2, \cdots), \cdots, (y_{n_p}^1, y_{n_p}^2, \cdots)\}$ is a codeword sequence of C_p, then, for any arbitrary sequence $\boldsymbol{z}(D) = (z_1(D), \cdots, z_{n_p}(D)) = \{(z_1^1, z_1^2, \cdots), \cdots, (z_{n_p}^1, z_{n_p}^2, \cdots)\}$ of binary n_p-tuples, it is satisfied that:

$$(\boldsymbol{z}(D) \oplus \boldsymbol{y}(D))\boldsymbol{H}_p^T(D) = \boldsymbol{z}(D)\boldsymbol{H}_p^T(D) = \boldsymbol{s}(D) \tag{5.14}$$

where $\boldsymbol{s}(D) = (s_1(D), \cdots, s_{r_p}(D)) = \{(s_1^1, s_1^2, \cdots), \cdots, (s_{r_p}^1, s_{r_p}^2, \cdots)\}$, $(r_p = n_p - k_p)$ is called the syndrome sequence, which identifies uniquely one of the $2^{J(n_p-k_p)}$ cosets[3] the code C_p and $\boldsymbol{z}(D)$ and $\boldsymbol{z}(D) \oplus \boldsymbol{y}(D)$ are 2 sequences belonging to the

[3] The set of all possible 2^{Jn_p} binary sequences of length J can be decomposed into $2^{J(n_p-k_p)}$ cosets, each having (in particular the code C_p) 2^{Jk_p} different binary sequences.

same coset. This property allows to have a smaller reduction in rate by transmitting a total of $n_u + r_p$ bits per 2 dimensions (instead of only n_u bits per 2 dimensions) in the following way. A sequence of binary r_p-tuples information bits will be identified sequentially as the syndrome sequence $\boldsymbol{s}(D)$. An initial representative sequence belonging to the coset identified by this syndrome sequence will be obtained by using any $(n_p - k_p) \times n_p$ (feedbackfree) left inverse $\boldsymbol{H}_p^{-T}(D)$ as $\boldsymbol{z}(D) = \boldsymbol{s}(D)\boldsymbol{H}_p^{-T}(D)$. For each codeword sequence $\boldsymbol{y}(D)$ (corresponding to one path in the trellis), the sequence $\boldsymbol{t}(D) = \boldsymbol{z}(D) + \boldsymbol{y}(D)$ will belong to the same coset. Let also $\boldsymbol{w}(D) = (w_1(D), \cdots, w_{n_u}(D)) = \{(w_1^1, w_1^2, \cdots), \cdots, (w_{n_u}^1, w_{n_u}^2, \cdots)\}$ be the sequence of binary n_u-tuples which determines the sequence of non-overlapping sets $\{\boldsymbol{A}_{\boldsymbol{m}_i}\}$. The sequence $\boldsymbol{t}(D)$ will determine a particular sequence of signal vectors belonging to the sequence of sets $\{\boldsymbol{A}_{\boldsymbol{m}_i}\}$ which are specified by the sequence $\boldsymbol{w}(D)$. Thus, in order to minimize the average power, the codeword sequence $\boldsymbol{y}(D)$ must be chosen optimally by finding the optimal path in the trellis corresponding to the minimum average power. This optimal path is found using the Viterbi decoding algorithm with branch metrics given by the squared norm of the signal vector $\boldsymbol{v}$ associated uniquely with each branch of the trellis defined by C_p. Let $\{\boldsymbol{v}'_{i_1}, \boldsymbol{v}'_{i_2}, \cdots, \boldsymbol{v}'_{i_J}\}$ be the sequence of signal vectors obtained at the decoder after applying Λ^1 and the modulo operation with Λ^3. Notice that in the case where there is no error decoding by Λ^1, this sequence will be identical to the sequence that was transmitted. First, the received sequence of signal vectors can be mapped to a sequence of J binary $(n_u + n_p)$-tuples giving a sequence of uncoded bits $\boldsymbol{w}'(D)$ and a sequence $\boldsymbol{t}'(D)$, which assuming that no decoding errors occurred, will be identical to the sequence $\boldsymbol{t}(D)$. In order to recover the r_p bits identifying uniquely the coset to which $\boldsymbol{t}(D)$ belongs to, the syndrome sequence is obtained by applying the syndrome-former, that is, $\boldsymbol{s}'(D) = \boldsymbol{t}'(D)\boldsymbol{H}_p^T(D)$. Since the syndrome-former can be chosen to be feedbackfree, in case of some error decoding event by Λ^1, the error propagation in the recovered syndrome sequence $\boldsymbol{s}'(D)$ will be only finite.

5.3.6 Shaping through a sequence of quantizers

Let $\Lambda^1/\Lambda^3/\Lambda^4$ be a lattice partition chain where $|\Lambda^1/\Lambda^3| = 2^{n_q}$ and $|\Lambda^3/\Lambda^4| = 2^{n_s}$. Let also the coset decomposition of Λ^3 be given by $\Lambda^3 = \Lambda^4_{c_1} \cup \Lambda^4_{c_2} \cup \cdots \cup \Lambda^4_{c_{2^{n_s}}}$ where $\Lambda^4_{c_1} = \Lambda^4$ and $\Lambda^4_{c_i}$ is the i-th coset. We denote by $Q_{\Lambda^4_{c_i}}$ the nearest neighbor quantizer whose reconstruction points are given by the coset $\Lambda^4_{c_i}$. Thus, all the quantizers are (congruent) shifted versions of each other. We can associate uniquely each of the 2^{n_s} binary n_s-tuples with a different coset. In this case, we associate with each message $\boldsymbol{m}$ one and only one signal vector $\boldsymbol{v}$ of the constellation composed by $\Lambda^1 \cap C_{\Lambda^3}(\boldsymbol{0})$, that is, no constellation expansion takes place, as opposed to the trellis precoding method. Let $\{\boldsymbol{m}_{i_1}, \boldsymbol{m}_{i_2}, \cdots, \boldsymbol{m}_{i_J}\}$ and $\{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_J\}$ again denote the sequence of J binary messages to be transmitted and the corresponding sequence of interference vectors. The shaping operation consists of choosing the best sequence $\{Q_{\Lambda^4_{c_{i_j}}}\}_{j=1}^{J}$ of quantizers such that the average power P per dimension is minimized:

$$Min_{\{Q_{\Lambda^4_{c_{i_j}}}\}_{j=1}^{J}} \quad P = \frac{1}{NJ}\sum_{j=1}^{J} \|(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j) - Q_{\Lambda^4_{c_{i_j}}}(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j)\|^2 \tag{5.15}$$

Notice that if we allow to choose any the 2^{n_s} quantizers at each time instant j, the minimization of the average power will be just the same as if we were using the baseline system explained in Section 5.3.3 with the sublattice being Λ^3. This is obvious because the union of the cosets is the whole lattice Λ^3, so finding the closest point to $(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j)$ in the union of the cosets is equivalent to finding the closest lattice point in Λ^3. Thus, in this particular case, the overall minimization for the J terms is equivalent to minimizing each term $\|(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j) - Q_{\Lambda^4_{c_{i_j}}}(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j)\|^2$ independently. On the other hand, since this system is equivalent to the baseline system, it will allow to transmit only n_q bits per 2 dimensions generating a uniform distribution of power in the Voronoi cell $C_{\Lambda^2}(\mathbf{0})$.

The goal we want to achieve is to increase the bit rate while generating a non-uniform distribution of power. The price to pay will be an increase in the peak power, in other words, the non-uniform distribution of power will have a somehow larger support. However, as we show next, this can be avoided by using peak power constraints in the trellis search. The idea is to constrain the choices of the quantizers through the whole sequence by using a k_s/n_s shaping convolutional code C_s where each path will correspond now to a particular sequence of quantizers $\{Q_{\Lambda^4_{c_{i_j}}}\}$. Thus, the shaping will be obtained by choosing among all the possible paths in the code C_s the path that gives the lowest average power. In this case, each branch of the trellis diagram of the code C_s will be associated with a particular quantizer and comming out from each state, there will be only 2^{k_s} quantizers to choose from.

Let $\boldsymbol{G}_s(D)$ be the $k_s \times n_s$ generator matrix of C_s and $\boldsymbol{H}_s^T(D)$ be a syndrome-former. Following the same arguments as for the trellis precoding idea, a sequence of binary r_s-tuples ($r_s = n_s - k_s$) information bits will be identified sequentially as the syndrome sequence $\boldsymbol{s}(D)$ and similarly an intial representative sequence $\boldsymbol{z}(D)$ will be obtained first using any $(n_s - k_s) \times n_s$ (feedbackfree) left inverse $\boldsymbol{H}_s^{-T}(D)$. However, this time, the information bits contained in the syndrome sequence $\boldsymbol{s}(D)$ will be transmitted through a constrained sequence of choices of quantizers. Each codeword sequence $\boldsymbol{y}(D)$ associated with each path in the trellis will give rise to a sequence $\boldsymbol{t}(D) = \boldsymbol{z}(D) + \boldsymbol{y}(D)$ of binary n_s-tuples. Then, this sequence $\boldsymbol{t}(D)$ will be mapped to a sequence of quantizers $\{Q_{\Lambda^4_{c_{i_j}}}\}$ using the previously agreeded mapping from binary n_s-tuples to the set of 2^{n_s} quantizers. Thus, in order to minimize the average power, the optimal (path) codeword sequence $\boldsymbol{y}(D)$ has to be found again using the Viterbi decoding algorithm with branch metrics given by the corresponding quantization error. More specifically, if a particular branch in the j-th step of the trellis is associated with a quantizer $Q_{\Lambda^4_{c_{i_j}}}$, its cost will be just $\|(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j) - Q_{\Lambda^4_{c_{i_j}}}(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j)\|^2$ where $\boldsymbol{v}_{i_j}$ is the signal vector to be transmitted at the time slot j.

At the decoder, the quantizer Q_{Λ^1} is applied to each received vector $\boldsymbol{y}_{i_j}$ at each time instant, obtaining, under the assumption that no decoding error occurred, a vector $\boldsymbol{b}_{i_j} = \boldsymbol{v}_{i_j} - Q_{\Lambda^4_{c_{i_j}}}(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j)$. Clearly, the n_q uncoded bits can be recovered without any problem by using the modulo operation with respect to Λ^3. We now show how we can obtain the r_s information bits per 2 dimensions. First, we need to recover the n_s bits identifying uniquely the quantizer $Q_{\Lambda^4_{c_{i_j}}}$ that was chosen (from the Viterbi algorithm) at the encoder side. Notice that the vector point $(-\boldsymbol{b}_{i_j}) = Q_{\Lambda^4_{c_{i_j}}}(\boldsymbol{v}_{i_j} - \boldsymbol{s}_j) - \boldsymbol{v}_{i_j}$ is actually closest to the coset that was chosen at the encoder, that is, $\Lambda^4_{c_{i_j}}$, so that we can find the correct coset by just obtaining first $Q_{\Lambda^3}(\boldsymbol{b}_{i_j})$ and then finding out to which of the 2^{n_s} cosets this quantized vector belongs to. This coset $\Lambda_{c_{i_j}}$ will be the closest to the vector $\boldsymbol{b}_{i_j}$ and it will give us the corresponding n_s bits. In this way, we will recover the whole sequence $\boldsymbol{t}(D)$ of binary n_s-tuples. The syndrome sequence which contains the the r_s information bits will be recovered by applying the syndrome-former, that is, $\boldsymbol{s}(D) = \boldsymbol{t}(D)\boldsymbol{H}_s(D)^T$. In the same way as in trellis precoding, in the case of some error decoding event by Λ^1, the error propagation will be finite.

In order to control the peak power in this scheme, one can simply impose peak constraints during the Viterbi search, that is, if a certain branch in the trellis chooses a coset $\Lambda^4_{c_{i_j}}$ which gives rise to a power that is above a certain preselected threshold, we associate an infinite cost to that branch so that this branch will never be selected.

An important feature of this shaping scheme is that it does not make any use of the constellation points (because there is no constellation expansion) in order to perform the shaping. From a complexity point of view, it is preferable to have a simple lattice Λ^4.

5.3.7 Shaping through a joint sequence of constellation points and quantizers

Let $\Lambda^1/\Lambda^2/\Lambda^3/\Lambda^4$ be a lattice partition chain where all the parameters are the same as in Sections 5.3.5 and 5.3.6. Now, we combine the two methods explained previously by allowing to choose jointly from a sequence of signal points and cosets.

Let T_p and T_s denote the trellis associated with the convolutional codes C_p and C_s respectively. Let T_p have states $v_1, \cdots, v_{2^{b_p}}$ and T_s have states $\omega_1, \cdots, \omega_{2^{b_s}}$. The combination of the two methods is performed through the tensor product $T_p \otimes T_s$ of the two trellises. The tensor product $T_p \otimes T_s$ is a trellis with $2^{b_p+b_s}$ states $v_n \otimes \omega_m$, $n = 1, \cdots, 2^{b_p}$, $m = 1, \cdots, 2^{b_s}$. There is a transition between states $v_n \otimes \omega_m$ and $v_t \otimes \omega_r$ in $T_p \otimes T_s$ if and only if there exist transitions between v_n and v_t in T_p and between ω_m and ω_r in T_s.

It is important to note that in the process of encoding and decoding, there is no need to actually implement the trellis $T_p \otimes T_s$, with $2^{b_p+b_s}$ states but one only needs to perform the operations for each time slot in two stages, one stage (sub-transition) using T_p and the other stage (sub-transition) using T_s. Thus, as usual, the survival path in the Viterbi decoding algorithm is obtained through a sequence of time slots, but in this case, in each time slot, the decoding is done succesively in two stages instead of one.

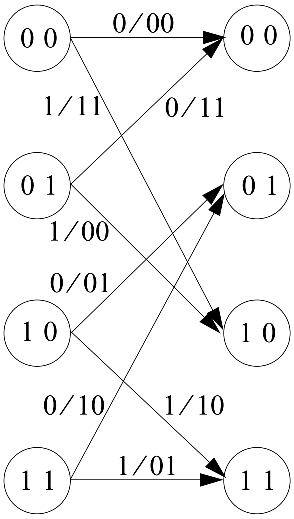


Figure 5.1: Trellis Diagram representing the four-state Ungerboeck code

Example Consider the case where $T_p = T_s = T_U$, where T_U denotes the four-state Ungerboeck's trellis code where $G(D) = [1 + D^2, 1 + D + D^2]$ and $H(D) =$

$[1 + D + D^2,\ 1 + D^2]$. The corresponding trellis diagram is illustrated in Fig. 5.1.

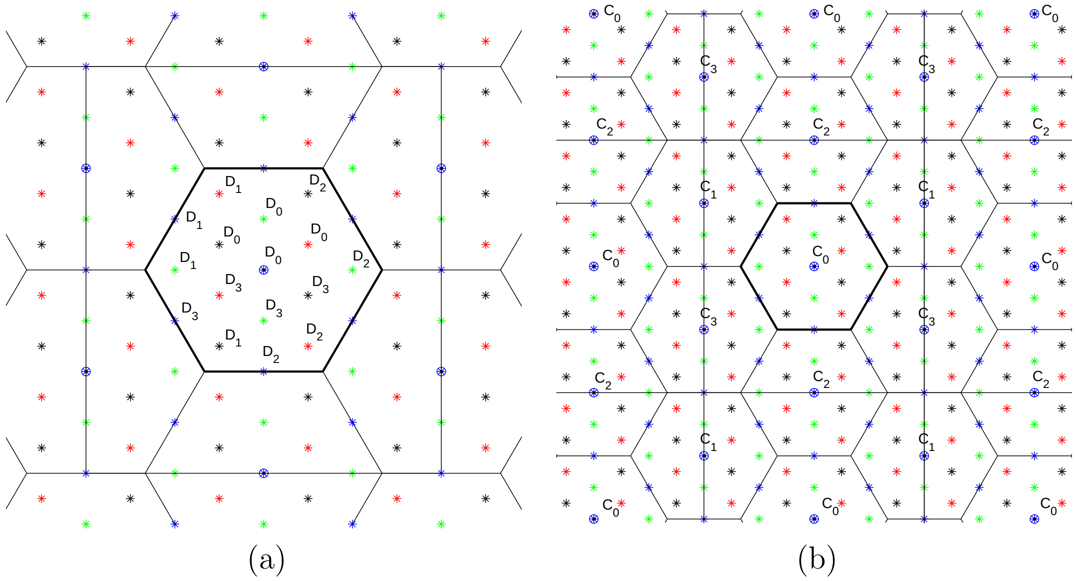


Figure 5.2: (a) Shaping through the fine lattice (channel code); (b) Shaping through the coarse lattice (source code).

Fig. 5.2 shows an example where $\Lambda^1 = A_2$, $\Lambda^2 = 2A_2$, $\Lambda^3 = 4A_2$ and Λ^4 is a rectangular lattice which is a sublattice of Λ^3 with index $|\Lambda^3/\Lambda^4| = 4$, where the Voronoi cells of Λ^3 and Λ^4 are shown. It can be seen that $C_{\Lambda^3}(\mathbf{0}) = |\Lambda^1/\Lambda^4|$ contains 16 constellation points (thus, the original rate in a baseline system would be 4 bits per $2D$ symbol), which are divided into 4 sets, each of them with 4 constellation points that we have called D_0, D_1, D_2 and D_3 Fig. 5.2(a). Each set is identified with a different colour. On the other hand, we have $|\Lambda^3/\Lambda^4| = 4$ cubic quantizers corresponding to the 4 cosets of the coarsest lattice Λ^4 with respect to Λ^3. The i-th coset is indicated in Fig. 5.2(b) with the label C_i.

For this example, we have that $k_p = k_s = 1$, $n_p = n_s = 2$ and $b_s = b_p = 2$ (4 states). The total transmitting rate per $2D$ symbol will be 4 (no shaping) - 1 (trellis precoding) + 1 (our approach) = 4 bits/symbol, which means that we keep the same rate as the original. The peak power can be controlled so that a gaussian-like power distribution is generated with support being the circle in which the hexagon in Fig. 5.2 is incribed, thus, without having an increase in the peak power. Therefore, the peak constraint in the Viterbi search will be the covering radius of this hexagon. Notice that the lattice Λ^4 has been choosen to be rectangular so that the complexity of the quantization operations at the encoder is lower than if we were quantizing with a more complicated lattice.

The complete evaluation of our designs has to be done by simulation and is the object of our ongoing work. Preliminary results show shaping gains over the corresponding baseline version of around 1dB and more depending on the complexity of the trellises and the decoding window size that are used [12].

5.4 Channel lattice decoding in fading channels

Previous work [19] on multidimensional modulation schemes for the fading channel show that good lattices can be obtained as rotated versions of lattices which are good for the Gaussian channel. Currently, there are not very efficient (low complexity) maximum-likelihood (minimum distance assuming equiprobable symbols) decoding algorithms for these lattices assuming a coherent (perfect channel state information) signal demodulation system. The fastest current decoding algorithm performs a full search within a certain sphere whose radius has to be chosen adaptively depending on the amount of fading and noise that are present [120]. The obvious advantage of this system with respect to a complete full search is that it never tests vectors which are outside the chosen sphere. However, there are several problems in this algorithm: a) heuristic rules are given in order to choose the smallest possible radius, which if not properly chosen, gives rise to a decoding failure, b) the complexity changes abruptly when the fading coefficients change significantly and c) most importantly, the average decoding complexity of this algorithm is very far from the complexity that takes place in the minimum distance decoding algorithms developed for the Gaussian channel.

The difficulty comes from the fact that the effect of the fading is to compress or enlarge independently each component of the lattice and the maximum-likelihood decoding now amounts to performing minimum distance decoding in this distorted space. The decoding algorithms developed for the Gaussian channel are based on certain algebraic properties of the original undistorted lattices. These algebraic properties are destroyed in the presence of fading and thus, these fast algorithms can not be used.

We think that it would be useful to make use of the decompositions we have found in [102] and which are described in Chapter 3 as an intermediate step to solve this problem. A surprising property that we have found in many of the decompositions considered in Chapter 3 is that the sum of lattices turns out to be equal to the union of the same lattices!. In order to clarify, let us consider a simple example where this property happens. As shown in Chapter 3, we can write the hexagonal lattice A_2 as $A_2 = \Lambda^1 \cap \Lambda^2 \cap \Lambda^3$, where $\Lambda^1, \cdots, \Lambda^3$ are rectangular lattices. Notice that taking duals and taking into account that A_2 and $\Lambda^1, \cdots, \Lambda^3$ are all modular lattices[4], it is easy to see that we can write A_2 also as the sum of three rectangular lattices $\Lambda'^1, \cdots, \Lambda'^3$ which are actually congruent to $\Lambda^1, \cdots, \Lambda^3$. However, the following additional properties can also be proved:

$$\begin{aligned}
&A_2 = \Lambda'^1 + \Lambda'^2 + \Lambda'^3 = \Lambda'^1 + \Lambda'^2 = \Lambda'^1 + \Lambda'^3 = \Lambda'^2 + \Lambda'^3 = \Lambda'^1 \cup \Lambda'^2 \cup \Lambda'^3 \\
&A_2\boldsymbol{RD} = \Lambda'^1\boldsymbol{RD} + \Lambda'^2\boldsymbol{RD} = \Lambda'^1\boldsymbol{RD} + \Lambda'^3\boldsymbol{RD} = \Lambda'^2\boldsymbol{RD} + \Lambda'^3\boldsymbol{RD} \\
&A_2\boldsymbol{RD} = \Lambda'^1\boldsymbol{RD} \cup \Lambda'^2\boldsymbol{RD} \cup \Lambda'^3\boldsymbol{RD}
\end{aligned} \tag{5.16}$$

where $\boldsymbol{R}$ represents a rotation and $\boldsymbol{D}$ is a diagonal matrix containing the fading coefficients, assuming that the fading is acting independently in each coordinate[5]. Thus, $A_2\boldsymbol{RD}$ represents a lattice which is obtained by first rotating A_2 and then

[4] As defined in Chapter 3, a modular lattice has the property that its dual lattice is geometrically similar to it.

[5] This can be achieved by using a signaling system with an interleaver of sufficiently large depth

applying the fading on the resulting lattice. The important property is that it is possible to decompose the lattice $A_2\boldsymbol{RD}$ as the *union* of lattices of the type $\Lambda\boldsymbol{RD}$ with Λ being a rectangular or cubic lattice. Similar properties hold also for many other decompositions that we have obtained, such as the decompositions for D_4 and E_8.

Therefore, since many of our decompositions can express each rotated lattice as a union of rotated cubic lattices (rotated QAM constellations) we can reduce in many cases the problem of decoding with respect to a faded complicated lattice to the problem of decoding lattices of the form $\Lambda\boldsymbol{RD}$ with Λ being a cubic lattice, which is a much more constrained lattice (hence, hopefully, easier to decode) than the faded original complicated lattice. Using the union decomposition in (5.16), the decoding with respect to the complicated will be obtained by decoding with respect to the lattices in the union and taking the best candidate, that is, the one that gives the smallest distance.

Therefore, we propose to study new fast decoding algorithms for the particular simplified case of having only cubic constellations, which we think is a problem that can be solved much more efficiently than in the case of having fading over a more complicated lattice. In the future, fast decoding algorithms for the case of non-coherent detection should be also elaborated using a similar simplifying intermediate step. Finally, this work may be also useful for decoding certain types of space-time codes (e.g. space-time codes obtained by having several transmit antennas using QAM signaling) [43] where the decoding has to be done with respect to an effective lattice which depends on the transfer matrix between the transmit and receive antennas.

5.5 Shiftable filter banks and their different applications

We can identify several issues to be explored:

1. It would be very convenient to try to find new designs (or optimization methods) of digital steerable filter banks which satisfy more closely the perfect reconstruction condition than the designs we have used in this work. Another possibility is to perform filtering in the frequency domain which allow filters to be designed analytically. On the other hand, we should use also filter banks with a smaller redundancy factor than $\frac{4J}{3}$ (J is the number of basic filters (orientations) because with the current design, even without oversteering, we have already a large amount of redundancy. The designs provided by Manduchi in [79] allow to decrease the redundancy factor but are based on a completely numerical approach using the SVD decomposition of matrices, which do not allow to have completely analytical expressions for the different elements (e.g., steering functions, etc...). A completely new

idea which is worth exploring is reduce the steerability requirement so that instead of allowing to perform steerability to any angle ϕ, we only require to satisfy steerability for a finite (sufficiently large) number of equispaced angles (discrete steerability). This relaxation of the steerability condition may allow for perfect reconstruction designs which can be manipulated in the polyphase domain (as it has been usually done traditionally in wavelet filter design), resulting in a set of filters that would perform better than the actual designs.

2. Some efficient algorithm should be found in order to achieve as good refinement (reduction of uncertainty in the angular domain) as possible between the different regions of uncertainty generated by the different sets of basic angles, when we add more and more orientations. Since it is not computationally feasible to consider all the possible combinations of basic angles, it should be interesting to find a (suboptimal) greedy approach with a much lower complexity and which gives good enough results of refinement.

3. It would be interesting to design embedded steerable transforms. For instance, given a steerable transform with J_1 basic angles, it would be interesting to design a steerable transform with $J_2 > J_1$ basic angles such that we can obtain the J_2 basic oriented subbands for the second steerable transform by using the J_1 basic subbands from the first steerable transform and

some additional information equivalent in size to $J_2 - J_1$ subbands in each corresponding level. This would be very useful in order to achieve succesive refinement in angle, which would be applicable for remote content-based image retrieval applications where it is necessary to have a certain scalability (e.g. the transmission bandwidth of the channel varies).

4. It should very interesting to study the design of filter banks which are steerable under the scaling group, that is, $g(\tau)\ f(x) = f(2^{-\tau}x)$. A filter bank like this would be very useful for instance in audio and speech applications (as well as in images), since it would allow to switch between different time-frequency tilings without having to refilter the signal with a different filter bank every time. With just one filtering operation, we would be able to obtain many different time-frequency tilings of the same signal, which would be very useful for both analysis and synthesis of signals. Although some theoretical analysis based on Lie theory [108, 110] has been developed in the context of functions steerable in scale, there does not exist yet a practical perfect reconstruction digital filter bank design which is steerable in scale.

5. Study the idea of using oversampled steerable transforms in the context of Multiple Description coding for images where each description would be

composed by a subset of orientations and/or scales and where the steerability property would be used explicitly to refine the information as the number of orientations and/or scales that are received increase.

6. As shown by the average percentages of correct retrieval in Chapter 4, it is necessary to treat differently textures that have clear predominant orientations and textures that are isotropic in energy. For textures with predominant orientations, the treatment we give in Chapter 4 seems to be appropiate. For isotropic textures such as ”sand”, the results show, as expected, that there is no real gain by using steerable transforms instead of regular wavelet transforms. Therefore, a different method should be used here. One possibility is to use a different similarity measurement which do not perform rotation of the features but takes also into account the stochastic nature of the texture, in addition to the energy-based features. For instance, one could use a joint stochastic model for the subbands where the subbands are fitted (using the Expectation-Maximization algorithm) with certain assumed probability density models and combine it with the energy based features. On the other hand, there are also some textures such as “bark” which are clearly inhomogeneous for which intutively, a global energy-based feature may not be the best option. One possibility for this is to subdivide further the texture blocks since in a local neighbourhood

the texture becomes more homogeneous. However, this will increase the number of features.

7. We have not addressed the *very important* problem of finding fast algorithms to obtain the optimal angle θ^* in (4.24) which is necessary to perform the similarity measurement. In the case of $J = 2$, it is easy to show that the angle θ^* can be found analytically because the matrix $\boldsymbol{R}(\theta)$ in (4.24) is a simple 2×2 rotation matrix in the plane and the minimization problem in (4.24) reduces to solving an equation of the type $ay^4 + by^2 + c = 0$ where a, b and c are known and depend only on the correlation matrix $\boldsymbol{C}$. For $J > 2$ the minimization problem in (4.24) does not reduce to an easily solvable equation. However, it can be easily shown that curve $D(\theta)$ can have at most J local extrema in the domain $[0\ \pi[$. In addition to this, since we know the steering functions, we have knowledge about the speed of change in the slope of the curve $D(\theta)$ across θ. We should use these facts in order to find a simple iterative search method (e.g. gradient-descendent search) to obtain a sufficiently good approximation to θ^*.

8. We have not used any training-based classifier for the retrieval process. Notice that in our experiments, given a query, we simply compared (aligning) the features of the query with the features of each of the texture samples in the database. In order to accelerate the retrieval process (reduce the

complexity in the retrieval), it is more desirable to have a Tree-Structure (TS) classifier which reduces very importantly the complexity of the system. This TS classifier is designed by training and in our case, it will trained using a sufficient large number of texture samples, all of them oriented at the *non-rotated* angle of 0 degrees. In the design of the TS, tree is grown in a greedy manner, where the splitting of the nodes is done according to some criterion, depending on which, a balanced or unbalanced tree is obtained. Each node of the tree will contain a representative set of features and given a query, the features of the query will be aligned with those of the node and a decision will be taking regarding which branch downwards is taken. Such a system (in particular, an unbalanced tree) although using wavelet energy-based features and not taking into account the rotation invariance problem, has been used by Hua and Ortega in [129, 88] where feature quantization is also taken into account. These TS schemes have been actually used previously for vector quantization applications. As a final comment, it would be convenient to test our schemes over a larger set of texture samples, including more classes of textures and more texture samples per texture. [93, 94].

9. In the presence of a remote content-based image retrieval application where it is necessary to quantize the features and thus, some error will be incurred,

it would be interesting to study whether it is better, from a recognition point of view and for a given total bit rate budget, to increase the number of quantized correlation matrices (obtained from different sets of basic angles) that are used per level, instead of using just one quantized correlation matrix (calculated from only one set of J basic angles) with the same total bit rate cost. Notice that in Chapter 4, we have analyzed the trade-off between oversteering and resolution (accuracy) at the transform coefficient level, but we have not analyzed this trade-off when we use features calculated over more than J angles and we quantize them.

5.6 Quantized overcomplete expansions as error correction codes over $\mathbb{R}$ or $\mathbb{C}$

Previous work by several researchers has pointed out that there are strong ties between digital signal processing and error correcting codes [18, 105, 53, 127]. systems but now over $GF(q)$. All the basic algebraic properties (e.g. calculation of determinants, Fourier transforms, inverses, convolution theorem) that are used in error correction codes such as Reed-Solomon codes hold just as well in any field and thus, are valid also over $\mathbb{R}$ and $\mathbb{C}$. This is because only the abstract structure of a field is used in all the analysis.

This is clearly seen by illustrating how a Reed-Solomon code can be defined in any field F though they have been used mostly in finite fields. A parallel argument for BCH codes can be made in exactly the same way but for the sake of simplicity we restrict to Reed-Solomon codes. Given a vector $\boldsymbol{x}$ composed by k information symbols in the generic field F of the code, a Reed-Solomon codeword is formed by appending $n-k$ new symbols called parity symbols. The codeword has length n and is constructed so that it is able to correct up to $t = \frac{n-k}{2}$ symbol errors. A Reed-Solomon code can be defined using the language of Fourier transforms. Let $\boldsymbol{c} = [c_0, \ldots, c_{k-1}]$ be a vector of length $n > k$ over a generic field F. The Discrete Fourier transform $\boldsymbol{C} = [C_0, \ldots, C_{n-1}]$ in F is given by:

$$C_j = \sum_{i=0}^{n-1} \omega^{ij} c_i \qquad j = 0, \ldots, n-1 \tag{5.17}$$

where ω is an nth root of unity in the field F. The t-error correction $(n, n-2t)$ $(k = n-2t)$ Reed-Solomon code is defined as the set of all vectors $\boldsymbol{c}$ such that $C_j = 0$ for $j = 0, \ldots, 2t-1$[6]. One way of finding the codewords is by performing encoding in the Fourier domain, that is, given a k-dimensional input vector of symbols, the corresponding codeword $\boldsymbol{c}$ is obtained as follows:

$$\boldsymbol{c} = \boldsymbol{W}_n^{-1} \boldsymbol{P} \boldsymbol{W}_k \boldsymbol{x} \tag{5.18}$$

[6] Actually, we can set to 0 any consecutive $2t$ transform coordinates

We first take a length-k DFT (denoted by $\boldsymbol{W}_k$) (invertible operation), then we perform a zero-padding of $2t$ consecutive positions, represented by a zero-padding matrix, and finally we project back to time domain by taking an IDFT of length n (denoted by $\boldsymbol{W}_n^{-1}$). Thus, a Reed-Solomon code can be viewed as a technique of digital signal processing that will protect a time-discrete waveform from impulsive noise or burst noise.

The only difference that appears (with respect to the case when F is a Galois field) when working over $\mathbb{R}$ or $\mathbb{C}$ is that prior to transmission through a channel, it is necessary to apply a quantizer Q over $\boldsymbol{c}$, obtaining $\hat{\boldsymbol{c}} = Q(\boldsymbol{c})$, and its representation in bits is sent through the channel. It is important to realize that the whole transform $\boldsymbol{W}_n^{-1}\boldsymbol{P}\boldsymbol{W}_k$ in (5.18) can be viewed as a *particular* overcomplete transform of redundancy $r = \frac{n}{k}$ and after quantization is included, what we have is a *particular quantized overcomplete expansion.* When F is a Galois field, the quantization is performed before applying error correction, thus there is a clear separation between source coding and channel coding. On the other hand, the real (or complex) case can be viewed more as a joint source-channel coding. If $\hat{\boldsymbol{c}}'$ is the received codeword, then there are two sources of distortion, namely, the quantization and the channel noise.

Regarding the decoding procedure of Reed-Solomon codes, it can also be shown that it actually consists of a spectral estimation problem, which is very usual in digital signal processing (e.g. deconvolution problem, spectral estimation, design of autoregressive filters, etc...). Algorithms for spectral estimation often require the solution of a Toeplitz system of equations and fast algorithms for finding the solution of a Toeplitz system of equations have been given by Levinson [76], Trench [114] and Durbin [48]. These algorithms are very well known within the field of digital signal processing. In the error-correction codes literature there are also algorithms, namely, the Berlekamp-Massey algorithm [17] and the Sugiyama method [103], which apply the Euclidean algorithm to find *also* the solution of Toeplitz systems of equations. These later algorithms have been widely used in finite fields but they are valid in any field F.

New ideas for research The overcomplete transform $\boldsymbol{T} = \boldsymbol{W}_n^{-1}\boldsymbol{P}\boldsymbol{W}_k$ in (5.18) is just a *very particular* overcomplete transform. In $\ell^2(\mathbb{Z})$ for instance, there are many different possible designs of oversampled filter banks. As an example, the first candidates that one can think of are Weyl-Heisenberg (Gabor) frames, which are basically windowed fourier exponentials. Complete designs of these overcomplete filter banks have been given by Cvetković in [38]. Using z-transform notation and also representing oversampled filter banks and signals in the polyphase domain, we have that one could in principle encode using:

$$\boldsymbol{c}_p(z) = [\boldsymbol{G}(z)_p]_{n_2 \times n_1} [\boldsymbol{P}_p(z)]_{n_1 \times n_1} [\boldsymbol{G}(z)_p]_{n_1 \times k} \boldsymbol{x}_p(z) \tag{5.19}$$

where $n_1 > k$, $n_2 < n_1$ and $n_2 > k$, introducing $n_2 - k$ redundant symbols and where $\boldsymbol{G}(z)_p$ is a synthesis oversampled filter bank (polyphase matrix) related to $\boldsymbol{H}(z)_p$, an analysis oversampled filter bank. $\boldsymbol{P}_p(z)$ will perform some operation to introduce structure in the redundancy as the zero-padding matrix. It should be interesting to study: a) how to design the best set of oversampled filter banks and the best matrix $\boldsymbol{P}_p(z)$ as a function of the statistics of the source and the channel in order to achieve good error correction performance. Another important issue to be investigated is the design of the quantizer which is applied to the coefficients of the resulting overcomplete transform. Its design should take into account both the statistics of the source and the conditions of the channel. Similarly, a decoding technique resembling and generalizing the spectral estimation method used for the DFT should be studied again taking into account the use of arbitrary oversampled filter banks.

These issues have not been investigated to date.

Bibliography

[1] Home page for Qhull. *http://www.geom.umn.edu/software/qhull, Geometry center.*

[2] B. Boots A. Okabe and K. Sugihara. *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams.* Wiley, NY, 1992.

[3] R. H. Bamberger. *The directional filter bank: a multirate filter bank for the directional decomposition of images.* Ph.D. thesis, Georgia Institute of Technology, 1990.

[4] R. H. Bamberger and M. J. T. Smith. A filter bank for the directional decomposition of images: Theory and design. *IEEE Trans. on Signal Proc.*, 40(4):882–893, 1992.

[5] C. B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Trans. Mathematical Software*, 22:469–483, 1996.

[6] E. S. Barnes and N.J.A. Sloane. The optimal lattice quantizer in three dimensions. *SIAM J. Algebraic Methods*, 4:30–41, 1983.

[7] Y. Be'ery and J. Snyders. Fast decoding of the Leech lattice. *IEEE J. Select. Areas Comm.*, 7:959–967, 1989.

[8] B. Beferull-Lozano and A. Ortega. Efficient quantization for overcomplete expansions in $\mathbb{R}^N$. *IEEE Trans. on Information Theory*, 49(1):129–150, January, 2003.

[9] B. Beferull-Lozano and A. Ortega. Coding techniques for oversampled steerable transforms. In *Int. Asilomar Conf. on Signals, Systems and Computers*, 1999.

[10] B. Beferull-Lozano and A. Ortega. Construction of low complexity regular quantizers for overcomplete expansions in $\mathbb{R}^N$. In *Proc. of Data Compression Conf.*, Snowbird, 2001.

[11] B. Beferull-Lozano and A. Ortega. Efficient quantization for overcomplete expansions in $\mathbb{R}^N$. In *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing*, Salt Lake City, 2001.

[12] B. Beferull-Lozano and S. Diggavi. Nested trellis codes and shaping for the transmitter side-information problem. In *Int. Symp. on Information Theory*, Yokohama, 2003.

[13] B. Beferull-Lozano, H. Xie and A. Ortega. Rotation-Invariant Features based on Steerable Transforms with an application to distributed image classification. In *Int. Conf. on Image processing*, Barcelona, 2003.

[14] B. Beferull-Lozano and A. Ortega. Rotation invariant content-based image retrieval with steerable filter banks. *IEEE Trans. on Image Processing.* In preparation.

[15] J. Belinfante and B. Kolman. A survey of Lie groups and Lie algebras with applications and computational methods. *SIAM*, 44(1), 1989.

[16] W. Bennett. Spectra of quantized signals. *Bell System Tech. Journal*, 27:446–472, 1948.

[17] E. R. Berlekamp. *Algebraic coding theory.* New York:Mc-Graw-Hill, 1968.

[18] R. E. Blahut. Signal processing and digital filtering. *Algebraic methods for signal processing and communications coding, C.S. Burrus ed., Springer Verlag*, 1992.

[19] J. Boutros, E. Viterbo, C. Rastello, and J. C. Belfiore. Good lattice constellations for both rayleigh fading and gaussian channels. *IEEE Trans. on Information Theory*, 42(2), 1996.

[20] G. Caire and S. Shamai (Shitz). On the achievable throughput of a multi-antenna gaussian broadcast channel. *Submitted to IEEE Trans. on Information Theory*, 2001.

[21] A. R. Calderbank, P. J. Cameron, W. M. Kantor, and J. J. Seidel. $\mathbb{Z}_4$-Kerdock codes, orthogonal spreads, and extremal Euclidean line-sets. *Proc. London Math. Soc.*, 75:436–480, 1997.

[22] A. R. Calderbank, R. H. Hardin, E. M. Rains, P. W. Shor, and N. J. A. Sloane. A group-theoretic framework for the construction of packings in Grassmannian spaces. *J. Algebraic Combin.*, 9:129–140, 1999.

[23] R. Carter, G. Segal, and I. Macdonald. *Lectures on Lie Groups and Lie Algebras.* Cambridge University Press, 1995.

[24] B. Chen and G. W. Wornell. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Trans. on Information Theory*, 47:1423–1443, 2001.

[25] J. Chou, S. S. Pradhan, and K. Ramchandran. Turbo-coded-trells-based constructions for data embedding:channel coding with side information. In *Int. Asilomar Conf. on Signals, Systems and Computers*, 1999.

[26] P. Chou, S. Mehrotra, and A. Wang. Multiple description decoding of overcomplete expansions using projections onto convex sets. In *Proc. of Data Compression Conf.*, pages 72–81, 1999.

[27] C. Chrysafis and A. Ortega. Efficient contex-based lossy wavelet image coding. In *Proc. of Data Compression Conf.*, 1997.

[28] A. Cohen. *An Introduction to the Lie theory of one-parameter groups; with applications to the solution of differential equations.* D.C. Heath & Co., 1911.

[29] A. S. Cohen and A. Lapidoth. The Gaussian watermarking game. *IEEE Trans. on Information Theory*, 48(6):1639–1667, 2002.

[30] J. H. Conway, E. M. Rains, and N.J.A. Sloane. On the existence of similar sublattices. *Canad. J. Math.*, 2000.

[31] J. H. Conway and N. J. A. Sloane. Voronoi regions of lattices, second moments of polytopes, and quantization. *IEEE Trans. on Information Theory*, 28:211–226, 1982.

[32] J. H. Conway and N. J. A. Sloane. The Coxeter-Todd lattice, the Mitchell group, and related sphere packings. *Math. Proc. Camb. Phil. Soc.*, 93:421–440, 1983.

[33] J. H. Conway and N. J. A. Sloane. The cell structures of certain lattices. *Miscellanea mathematica, P. Hilton, F. Hirzebruch and R. Remmert, editors, Springer-Verlag*, 28:71–107, 1991.

[34] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups.* Springer-Verlag, 1998.

[35] J. H. Conway and N.J.A. Sloane. Fast quantizing and decoding algorithms for lattice quantizers and codes. *IEEE Trans. on Information Theory*, 28:227–232, 1982.

[36] M. Costa. Writing on dirty paper. *IEEE Trans. on Information Theory*, 29(3):439–441, 1983.

[37] H. S. M. Coxeter. *Regular Polytopes.* Dover, NY, 3rd, 1973.

[38] Z. Cvetković. *Overcomplete Expansions for Digital Signal Processing.* Ph.D. Thesis, Univ. California, Berkeley. Available as Univ. California, Berkeley, Electron. Res. Lab. Memo. No. UCB/ERL M95/114, 1995.

[39] Z. Cvetković. Source coding with quantized redundant expansions: accuracy and reconstruccion. In *Proc. of Data Compression Conf.*, pages 344–353, Snowbird, 1999.

[40] Z. Cvetković. Properties of redundant expansions under additive degradation and quantization. *Submitted to IEEE Trans. on Info. Theory*, 2001.

[41] Z. Cvetković and I. Daubechies. Single-bit oversampled A/D conversion with exponential accuracy in the bit-rate. In *Proc. of Data Compression Conf.*, pages 343–352, 2000.

[42] Z. Cvetković and M. Vetterli. On simple oversampled A/D conversion in $L^2(\mathbb{R}^2)$. *IEEE Trans. on Information Theory*, 47(1):146–154, 2001.

[43] O. Damen, A. Chkeif, and Jean-Claude Belfiore. Lattice code decoder for space-time codes. *IEEE Communications Letters*, 4(5):161–163, 2000.

[44] I. Daubechies. *Ten lectures on wavelets.* SIAM, 1992.

[45] S. N. Diggavi, N. J. A. Sloane, and V. A. Vaishampayan. Design of asymmetric multiple description lattice vector quantizers. In *Proc. of Data Compression Conf.*, pages 490–499, Snowbird, 2000.

[46] E. Dubois. The sampling and reconstruction of time-varying imagery with application in video systems. *Proc. of the IEEE*, 73(4), 1985.

[47] C. C. MacDuffee. *The theory of Matrixes.* New York: Chelsea, 1946.

[48] J. Durbin. The fitting of time-series models. *Rev. Int. Statist. Inst.*, 23:233–244, 1960.

[49] M. V. Eyuboglu and G. D. Forney. Trellis precoding: combined coding,precoding and shaping for intersymbol interference channels. *IEEE Trans. on Information Theory*, 38(2):301–313, 1992.

[50] G. D. Forney. Trellis shaping. *IEEE Trans. on Information Theory*, 38(2):281–300, 1992.

[51] J. Franca, A. Petraglia, and S. K. Mitra. Multirate analog-digital systems for signal processing and conversion. Proc. of the IEEE, 85(2):469–479, 1997.

[52] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 13(9):891–906, 1991.

[53] A. Gabay, O. Rioul, and P. Duhamel. Real number transform and convolutional codes. In *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing*, 1981.

[54] A. V. Geramita and J. Seberry. *Orthogonal designs: quadratic forms and Hadamard matrices.* Marcel Dekker Inc., 1979.

[55] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer, Boston, 1992.

[56] I. J. Good and R. A. Gaskins. Second moments of convex polytopes. *Numerical Math*, 16:343–359, 1971.

[57] V. Goyal and J. Kovacević. Optimal multiple description coding of gaussian vectors. In *Proc. of Data Compression Conf.*, pages 388–397, 1998.

[58] V. K. Goyal and J. Kovacevic. Quantized frame expansions with erasures. *Applied and Comp. Harmonic Analysis*, To appear in 2001.

[59] V. K. Goyal, J. Kovacević, and M. Vetterli. Quantized frame expansions as source-channel codes for erasures using tight frame expansions. In *Proc. of Data Compression Conf.*, pages 326–335, Snowbird, 1998.

[60] V. K. Goyal, M. Vetterli, and N. T. Thao. Quantized overcompleted expansions in $\mathbb{R}^n$: Analysis, synthesis and algorithms. *IEEE Trans. on Information Theory*, 44(1):16–31, 1998.

[61] R. M. Gray. *Source Coding Theory.* Kluwer, Boston, 1990.

[62] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. on Information Theory*, 44:2325–2383, 1998.

[63] R. M. Gray and T. G. Stockham. Dithered quantizers. *IEEE Trans. on Information Theory*, 39(3):805–812, 1993.

[64] R. H. Hardin, N. J. A. Sloane, and Warren D. Smith. Tables of spherical codes. *http://www.research.att.com/∼njas/packings/.*

[65] S. Hein and A. Zakhor. Reconstruction of oversampled bandlimited signals from sigma delta encoded binary sequences. *IEEE Trans. on Signal Proc.*, 42(4):799–811, 1994.

[66] S. Hein and A. Zakhor. Theoretical and numerical aspects of an SVD-based approach to bandlimiting finite extent sequences. *IEEE Trans. on Signal Proc.*, 42(5):1227–1230, 1994.

[67] Y. Hel-Or and P. Teo. Canonical decomposition of steerable functions. *Int. Conf. on Computer vision and pattern recognition*, pages 809–816, 1996.

[68] Y. Hel-Or and P. Teo. Canonical decomposition of steerable functions. *J. of Mathematical imaging and vision*, 9(1): 83–95, 1998.

[69] R. Herman. *Lie Groups for Physicists.* W. A. Benjamin, Inc., 1966.

[70] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. of Physiology*, 160:106–154, 1962.

[71] D. Hubel and T. Wiesel. Brain mechanisms of vision. *Scientific American*, 1979.

[72] N. Kashyap and D. L. Neuhoff. On quantization with the Weaire-Phelan partition. *IEEE Trans. Inform. Theory*, 2002.

[73] A. Kirillov. *Elements of the Theory of Representations.* Springer-Verlag, 1976.

[74] Y. Kitaoka. *Arithmetic of Quadratic forms.* Cambridge, University Press, 1993.

[75] A. Knapp. *Representation theory of semisimple groups.* Princeton University Press, 1986.

[76] N. Levinson. The Wiener RMS error criterion in filter design and prediction. *J. Math. Phys.*, 25:261–278, 1947.

[77] H. A. Loeliger. Averaging bounds for lattices and linear codes. *IEEE Trans. on Information Theory*, 43:1767–1773, 1997.

[78] R. Manduchi G. M. Cortelazzo and G. A. Mian, Multistage sampling structure conversion of video signals. *IEEE Trans. on Circ. and Syst. for Video Tech.*, 3:325–340, 1993.

[79] R. Manduchi. Pyramidal implementation of deformable kernels. In *Int. Conf. on Image processing*, pages 378–381, 1995.

[80] J. E. Mazo and J. Salz. On the transmitted power in generalized partial response. *IEEE Trans. on Comm.*, 24:348–352, 1992.

[81] G. Nebe, E. M. Rains, and N. J. A. Sloane. The invariants of the Clifford groups. *Designs, Codes and Cryptography*, 24:99–121, 2001.

[82] G. Nebe, E. M. Rains, and N. J. A. Sloane. A simple construction for the Barnes-Wall lattices. *The Forney Festschrift, To appear*, 2002.

[83] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* Wiley-Interscience, 1999.

[84] N. T. Thao and M. Vetterli. Deterministic analysis of oversampled A/D conversion and decoding improvement based on consistent estimates. *IEEE Trans. on Signal Proc.*, 42(3):519–531, 1994.

[85] University of Southern California. Rotated textures. *Signal and Image Processing Institute*, http://sipi.usc.edu/services/database/Database.html.

[86] P. Olver. *Equivalence, Invariants, and Symmetry.* Cambridge University Press, 1995.

[87] O. T. O'Meara. *Introduction to Quadratic Forms.* Springer-Verlag, 1971.

[88] A. Ortega, B. Beferull-Lozano, N. Srinivasamurthy, and H. Xie. Compression for recognition and content-based retrieval. In *Proc. of Eusipco*, 2000.

[89] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. *Image and Vision Computing*, (10):663–672, 1992.

[90] P. Perona. Deformable kernels for early vision. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 5(17):488–499, 1995.

[91] A. Petraglia and S. K. Mitra. High-speed A/D conversion incorporating a QMF bank. IEEE Trans. on Instrumentation and Measurement, 41(3):427–431, 1992.

[92] S. Rangan and V. K. Goyal. Recursive consistent estimation with bounded noise. *IEEE Trans. on Information Theory*, 47(1), 2001.

[93] E. A. Riskin. *Variable rate vector quantization of images.* Ph.D. Thesis, Stanford University, 1990.

[94] E. A. Riskin. Optimal bit allocation via the generalized BFOS algorithm. *IEEE Trans. on Information Theory*, 37:400–402, 1991.

[95] A. Sagle and R. Walde. *Introduction to Lie groups and Lie algebras.* Academic Press, 1973.

[96] A. Said and W.A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circ. and Syst. for Video Technology*, 6(3):243–250, 1996.

[97] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Proc.*, 41(12):3445–3462, 1993.

[98] E. Simoncelli. Design of multi-dimensional derivative filters. In *Int. Conf. on Image processing*, volume 1, pages 790–794, 1994.

[99] E. Simoncelli, W. Freeman, and D. Heeger. Shiftable multiscale transforms. *IEEE Trans. on Information Theory*, 38(2):587–602, 1992.

[100] E.P. Simoncelli, W. Freeman, E. Adelson, and D. Heeger. Shiftable multiscale transforms. *IEEE Trans. on Information Theory*, 2(38):587–607, 1992.

[101] E.P. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. *International Conference on Image Processing*, 1998.

[102] N. J. A. Sloane and B. Beferull-Lozano. Quantizing using lattice Intersections. *To appear in Journal of Discrete and Computational Geometry*, January 2003.

[103] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. A method for solving key equations for decoding Goppa codes. *Inform. Contr.*, 27:87–99, 1975.

[104] D. F. Swayne, D. Cook, and A. Buja. XGobi: interactive dynamic graphics in the X window system with a link to S. *ASA Proceedings Section on Statistical Graphics, Amer. Stat. Assoc, http://www.research.att.com/areas/stat/xgobi/*, pages 1–8, 1991.

[105] Jr. T. G. Marshall. Real number transform and convolutional codes. In *24th Midwest Symp. on Circuits and Systems*, Albuquerque, 1981.

[106] J. Talman. *Special functions; a group theoretic approach.* W. A. Benjamin, 1968.

[107] P. Teo and Y. Hel-Or. A computational approach to steerable functions. *Int. Conf. on computer vision and pattern recognition*, , pages 313–318, 1997.

[108] P. Teo and Y. Hel-Or. Design of multi-parameter steerable functions using cascade basis reduction. In *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 21(6):552–556, 1998.

[109] P. Teo and Y. Hel-Or. Design of multi-parameter steerable functions using cascade basis reduction. In *Int. Conf. on computer vision*, pages 187–192, 1998.

[110] P. Teo and Y. Hel-Or. A computational group-theoretic approach to steerable functions. *Pattern Recognition*, 19(1):7–17, 1998.

[111] N. T. Thao and M. Vetterli. Reduction of the MSE in r-times oversampled A/D conversion from O($1/r$) to O($1/r^2$). *IEEE Trans. on Signal Proc.*, 42(1):200–203, 1994.

[112] N. T. Thao and M. Vetterli. Lower bound on the mean-squared error in oversampled quantization of periodic signals using vector quantization analysis. *IEEE Trans. on Information Theory*, 42(2):469–479, 1996.

[113] L. Fejes Tóth. Sur la représentation d'une population infinie par une nombre fini d'éléments. *Acta Mathematica Academiae Scientiarum Hungaricae*, 10:299–304, 1959.

[114] W. F. Trench. An algorithm for the inversion of finite Toeplitz matrices. *J. SIAM*, pages 512–522.

[115] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto. Multiple description vector quantization with lattice codebooks: design and analysis. *IEEE Trans. on Information Theory*, 47:1718–1734, 2001.

[116] A. Vardy. Even more efficient bounded-distance decoding of the hexacode, the Golay code, and the Leech lattice. *IEEE Trans. on Information Theory*, 41:1495–1499, 1995.

[117] A. Vardy and Y. Be'ery. Maximum-likelihood decoding of the Leech lattice. *IEEE Trans. on Information Theory*, 39:1435–1444, 1993.

[118] M. Vetterli. Multidimensional subband coding: some theory and algorithms. *Signal Proc.*, 6(2):97–112, 1984.

[119] M. Vidyasagar. *Control System Synthesis: A Factorization approach.* Cambridge: The MIT Press, 1985.

[120] E. Viterbo and J. Boutros. A universal lattice code decoder for fading channels. *IEEE Trans. on Information Theory*, pages 1639–1642, 1999.

[121] J. P. Ward. *Quaternions and Cayley Numbers.* Kluwer Academic Publishers, 1997.

[122] Lee-Fang Wei. Trellis-coded modulation with multidimensional constellations. *IEEE Trans. on Information Theory*, 33(4):483–501, 1987.

[123] Lee-Fang Wei. Coded M-DPSK with built-in time diversity for fading channels. *IEEE Trans. on Information Theory*, 39(6):1820–1839, 1993.

[124] C. Weibel. *An introduction to algebraic K-theory.* Available online, http://www.rutgers.edu/∼Weibel, 2001.

[125] A. F. Wells. *Three-Dimensional Nets and Polyhedra.* Wiley,NY, 1977.

[126] R. Wilson and H. Knutsson. Uncertainty and inference in the visual system. *IEEE Trans. on systems, man and cybernetics*, 18(2):305–312, 1988.

[127] J. K. Wolf. Redundancy, the discrete Fourier transform, and impulse noise cancellation. *IEEE Trans. on Comm.*, COM-31:458–461, 1983.

[128] J.W. Woods. *Subband image processing.* Kluwer Academic Publishers, Boston, MA, 1991.

[129] H. Xie and A. Ortega. Entropy-and-complexity-constrained classified quantizer design for distributed image classification. In *Proc. of Multimedia Signal Processing (MMSP)*, 2002.

[130] D. Youla. *Mathematical theory of image restoration: the method of convex projections.* Image Recovery Theory and Application, 1987.

[131] D. C. Youla. *Mathematical theory of image restoration: the method of convex projections.* Image Recovery Theory and Application, 1987.

[132] W. Yu and J. M. Cioffi. Sum capacity of a gaussian vector broadcast channel. *Submitted to IEEE Trans. on Information Theory*, 2002.

[133] W. Yu, G. Ginis, and J. Cioffi. Distributed multiuser power control for digital subscriber lines. *To appear in IEEE Journal on Selected Areas of Communications*, 2002.

[134] Wei Yu. The gaussian watermarking game. *Private Communication*, pages –, 2000.

[135] P. L. Zador. Development and evaluation of procedures for quantizing multivariate distributions. In *Ph.D. Dissertation*, pages 378–381, Stanford Univ., 1963.

[136] P. L. Zador. Asymptotic quantization error of continuous signals and their quantization dimension. *IEEE Trans. on Information Theory*, 28:139–148, 1982.

[137] R. Zamir, S. Shamai (Shitz), and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Trans. on Information Theory*, 48(6):1250–1276, 2002.

Appendix A

A.1 Proof of Fact 1

Since Λ_s is a sublattice of Λ^1, Λ_s is a subgroup of the additive group Λ^1, and the result follows directly by group theory. The periodicity is determined by the subgroup and therefore the minimal periodic unit is given by the tiling contained in $C_o^{\Lambda_s}$, the fundamental polytope associated with the sublattice Λ_s. Since the subgroup structure is true for any dimension N, the periodicity property is also true for any dimension N $\square$.

A.2 Proof of Lemma 1

Let $\boldsymbol{M}_{\Lambda^j} = \boldsymbol{A}_{\Lambda^j}\boldsymbol{M}_{\Lambda^1}$ and consider the matrix $\boldsymbol{A}_{\Lambda^j}$ given by:

$$\boldsymbol{A}_{\Lambda^j} = \begin{pmatrix} \frac{1}{d_1^j} & 0 \\ 0 & \frac{1}{d_2^j} \end{pmatrix} \begin{pmatrix} k_{11}^j & k_{12}^j \\ -k_{21}^j & k_{22}^j \end{pmatrix} \tag{A.1}$$

whose inverse is equal to:

$$(\boldsymbol{A}_{\Lambda^j})^{-1} = \frac{1}{k_{11}^j k_{22}^j + k_{12}^j k_{21}^j} \begin{pmatrix} k_{22}^j d_1^j & k_{12}^j d_2^j \\ -k_{21}^j d_1^j & k_{11}^j d_2^j \end{pmatrix} = \frac{1}{D^j} \begin{pmatrix} t_{11}^j & t_{12}^j \\ t_{21}^j & t_{22}^j \end{pmatrix} \tag{A.2}$$

where $t_{lm}^j \in \mathbb{Z}$ and $D^j \in \mathbb{Z}_+$ is the denominator that is left after all the common factors have been canceled out. For each j we define the lattice $\Lambda^{j'}$ with generator matrix given by:

$$\boldsymbol{M}_{\Lambda^{j'}} = D^j(\boldsymbol{A}_{\Lambda^j})^{-1}\boldsymbol{M}_{\Lambda^j} = D^j(\boldsymbol{A}_{\Lambda^j})^{-1}\boldsymbol{A}_{\Lambda^j}\boldsymbol{M}_{\Lambda^1} = D^j\boldsymbol{M}_{\Lambda^1} \tag{A.3}$$

Notice that $\Lambda^{j'} \subset \Lambda^j$ is a sublattice of Λ^j because the matrix $D^j(\boldsymbol{A}_{\Lambda^j})^{-1}$ has integer entries. After calculating $\boldsymbol{M}_{\Lambda^{j'}}$ $\forall$ $j = 1, \cdots, r$, we define D as $D = l.c.m(D^1, D^2, \cdots, D^r)$ and the lattice Λ^o with generator matrix $\boldsymbol{M}_{\Lambda^o} = D\boldsymbol{M}_{\Lambda^1}$, which means that Λ^o is an integer scaling of Λ^1. Thus, we have that $\Lambda^o \subset \Lambda^{j'} \subset \Lambda^j \subset \Lambda^1$, $\forall\ j = 1, \cdots, r$. This implies clearly that $\Lambda^o \subset (\Lambda^1 \cap \Lambda^2 \cap \cdots \cap \Lambda^r)$ and therefore, Λ^o is a sublattice of the coincidence site lattice Λ^{CSL} ■.

A.3 Proof of Lemma 2

Since Λ^{CSL} is the finest sublattice of all the lattices Λ^j, $j = 1, \cdots, r$, if we consider any cell C_i^{CSL}, the relative positions of the lattice points $\{\boldsymbol{v}_i^j\}$ (vertices of the cells associated with Λ^j) for each lattice Λ^j, which are inside the cell C_i^{CSL}, these positions are always the same independently of which cell C_i^{CSL} is chosen. This immediately implies that the structure of the resulting EVQ is a periodic repetition of the structure of cells that is inside the fundamental polytope C_o^{CSL} of the CSL ■.

A.4 Proof of Fact 2

The proof follows in a straightforward manner by direct calculation from the definition of sublattice, which implies that

$$\begin{pmatrix} c_1\Delta_1^1 & 0 \\ 0 & c_2\Delta_2^1 \end{pmatrix} \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix} = \begin{pmatrix} k_{11}\Delta_1^1 & k_{12}\Delta_2^1 \\ -k_{21}\Delta_1^1 & k_{22}\Delta_2^1 \end{pmatrix} \tag{A.4}$$

Hence, a set of sufficient conditions is given by:

$$\begin{array}{l} c_1\Delta_1^1[\cos(\theta), \sin(\theta)] = [k_{11}\Delta_1^1, k_{12}\Delta_2^1] \\ c_2\Delta_2^1[-\sin(\theta), \cos(\theta)] = [-k_{21}\Delta_1^1, k_{22}\Delta_2^1], \qquad k_{11}, k_{12}, k_{21}, k_{22} \in \mathbb{Z} \end{array} \tag{A.5}$$

If we use the variable $\beta = \frac{\Delta_2^1}{\Delta_1^1}$ and simplify the previous equations, we get that:

$$c_1 \cos(\theta) = k_{11} \tag{A.6}$$

$$c_1 \sin(\theta) = \beta k_{12} \tag{A.7}$$

$$-c_2\beta \sin(\theta) = -k_{21} \tag{A.8}$$

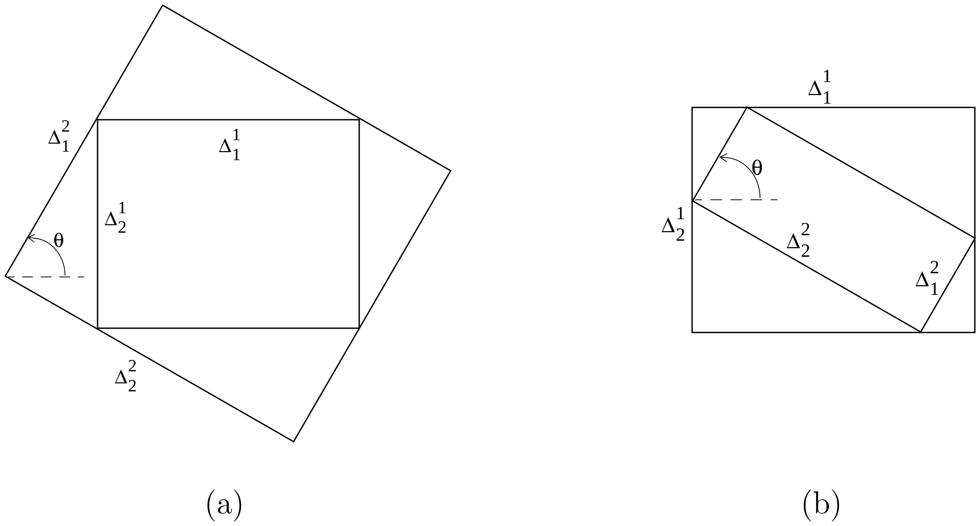
$$c_2 \cos(\theta) = k_{22} \tag{A.9}$$

Without loss of generality, we consider the case $0 < \theta < \frac{\pi}{2}$. This constrains the signs of all the integers k_{11}, k_{12}, k_{21} and k_{22} to be positive. Solving the previous equations for β and θ results in:

$$\beta = \sqrt{\tfrac{k_{11}k_{21}}{k_{12}k_{22}}}, \qquad \tan(\theta) = \sqrt{\tfrac{k_{12}k_{21}}{k_{11}k_{22}}} \tag{A.10}$$

The values for c_1 and c_2 follow from using (A.6) and (A.7).

A.5 Geometric constraints on the stepsizes



(a) (b)

Figure A.1: 2 Limiting cases for the stepsizes Δ_1^2 and Δ_2^2 of the quantizer Q^2.

Let us consider for simplicity the case of $r = 2$. The approach we have followed is to constrain the possible stepsizes Δ_1^2, Δ_2^2 to have values between the 2 limiting cases that happen when the Voronoi region of one quantizer is totally inside of a Voronoi cell of the other quantizer, as shown in Figure A.1. These 2 limiting cases establish upper and lower bounds for the pair (d_1^2, d_2^2) such that all pairs in between will satisfy the property. By using elemental trigonometry, we can calculate upper and lower bounds for the pair (d_1^2, d_2^2).
From Fig. A.1(a), we get that:

$$\begin{array}{l} \Delta_2^1 \sin(\theta) + \Delta_1^1 \cos(\theta) \geq \Delta_1^2 \Longrightarrow d_1^2 \geq \frac{k_{11}^2 k_{22}^2 + k_{12}^2 k_{21}^2}{k_{21}^2 + k_{22}^2} \\ \Delta_1^1 \sin(\theta) + \Delta_2^1 \cos(\theta) \geq \Delta_2^2 \Longrightarrow d_2^2 \geq \frac{k_{11}^2 k_{22}^2 + k_{12}^2 k_{21}^2}{k_{11}^2 + k_{12}^2} \end{array} \qquad (A.11)$$

which gives lower bounds for d_1^2 and d_2^2.

In the same way, from Fig. A.1(b), we get that:

$$\begin{array}{l} \Delta_1^2 \sin(\theta) + \Delta_2^2 \cos(\theta) \geq \Delta_2^1 \Longrightarrow \frac{k_{12}^2}{d_1^2} + \frac{k_{22}^2}{d_2^2} \geq 1 \\ \Delta_2^2 \sin(\theta) + \Delta_1^2 \cos(\theta) \geq \Delta_1^1 \Longrightarrow \frac{k_{11}^2}{d_1^2} + \frac{k_{21}^2}{d_2^2} \geq 1 \end{array} \qquad (A.12)$$

which gives upper bounds for d_1^2 and d_2^2.

For instance, in Example 1, the pairs (d_1^2, d_2^2) are constrained by:

$$\begin{array}{ll} d_1^2 \geq \frac{7}{4}, & d_2^2 \geq \frac{7}{3} \\ 2d_2^2 + d_1^2 \geq d_1^2 d_2^2, & 3d_1^2 + d_2^2 \geq d_1^2 d_2^2 \end{array} \tag{A.13}$$

which limits the possible values of (d_1^2, d_2^2) to $\{(2,3),(2,4),(2,5),(2,6),(2,7),(3,3)\}$.

A.6 Proof of Lemma 3

By adding and subtracting (2.39) and (2.40), and doing the same for (2.41) and (2.42), we get the following equivalent set of equations:

$$\begin{aligned} \Delta_1^2 \cos(\theta) = \frac{q_1 + q_2}{2}\Delta_1^1 &= q_1'\Delta_1^1 && \text{(A.14)} \\ \Delta_2^2 \sin(\theta) = \frac{q_2 - q_1}{2}\Delta_1^1 &= q_2'\Delta_1^1 && \text{(A.15)} \\ \Delta_1^2 \sin(\theta) = \frac{q_3 + q_4}{2}\Delta_2^1 &= q_3'\Delta_2^1 && \text{(A.16)} \\ \Delta_2^2 \cos(\theta) = \frac{q_3 - q_4}{2}\Delta_2^1 &= q_4'\Delta_2^1 && \text{(A.17)} \\ \text{with} \quad q_1', q_2', q_3', q_4' &\in \mathcal{Q} \end{aligned}$$

Assume that (A.14), (A.15), (A.16) and (A.17) are satisfied. Manipulating these equations, we get:

$$\text{Dividing (A.14) and (A.15),} \quad \tan(\theta) = \tfrac{\Delta_1^2}{\Delta_2^2}\tfrac{q_2'}{q_1'} \tag{A.18}$$

$$\text{Dividing (A.16) and (A.17),} \quad \tan(\theta) = \tfrac{\Delta_2^2}{\Delta_1^2}\tfrac{q_3'}{q_4'} \tag{A.19}$$

$$\text{Dividing (A.14) and (A.17),} \quad \tfrac{\Delta_1^2}{\Delta_2^2} = \tfrac{\Delta_1^1}{\Delta_2^1}\tfrac{q_1'}{q_4'} \tag{A.20}$$

$$\text{Dividing (A.15) and (A.16),} \quad \tfrac{\Delta_1^2}{\Delta_2^2} = \tfrac{\Delta_2^1}{\Delta_1^1}\tfrac{q_1'}{q_4'} \tag{A.21}$$

Solving these equations and expressing all the stepsizes in terms of Δ_1^1 we obtain:

$$\tan(\theta) = \sqrt{\frac{q_2' q_3'}{q_1' q_4'}} \tag{A.22}$$

$$\Delta_2^1 = \beta\Delta_1^1 = \sqrt{\frac{q_1' q_2'}{q_3' q_4'}}\Delta_1^1 \tag{A.23}$$

$$\Delta_1^2 = \frac{q_1'}{\cos(\theta)}\Delta_1^1 \tag{A.24}$$

$$\Delta_2^2 = \sqrt{\frac{q_1'q_3'}{q_2'q_4'}}\Delta_1^2 = \frac{q_4'}{\cos(\theta)}\sqrt{\frac{q_1'q_2'}{q_3'q_4'}}\Delta_1^1 = \frac{q_4'}{\cos(\theta)}\beta\Delta_1^1 \tag{A.25}$$

If we compare (A.22), (A.23), (A.24) and (A.25) with (2.19) and (2.20), we have obtained exactly the same equations with $q_1' = \frac{k_{11}^2}{d_1^2}$, $q_2' = \frac{k_{21}^2}{d_2^2}$, $q_3' = \frac{k_{12}^2}{d_1^2}$ and $q_4' = \frac{k_{22}^2}{d_2^2}$. Since the final set of equations is equivalent to the first 4 equations (2.39), (2.40), (2.41) and (2.42), it is clear that this Lemma is also true in the other direction ■.

A.7 Proof of Theorem 2

Without loss of generality, we can assume a quantizer Q^1 associated with a lattice Λ^1 where $\boldsymbol{F}^1 = \boldsymbol{I}_{2\text{x}2}$ and $\boldsymbol{M}_{\Lambda^1} = diag[\Delta_1^1, \Delta_2^1]$. A general quantizer Q^2 can be associated with a lattice Λ^2. We denote by $\boldsymbol{x}|_i$ the components of $\boldsymbol{x}$ expressed in the basis $\{\boldsymbol{\varphi}_1^i, \boldsymbol{\varphi}_2^i\}$, $i = 1, 2$, where $i = 1$ indicates, without loss of generality, the natural basis. In order to find an inconsistent cell, we consider the vertices of Λ^2. Any vertex can be written as:

$$\boldsymbol{\omega} = k_1(\Delta_1^2\boldsymbol{\varphi}_1^2 + \Delta_2^2\boldsymbol{\varphi}_2^2) + k_2(\Delta_1^2\boldsymbol{\varphi}_1^2 - \Delta_2^2\boldsymbol{\varphi}_2^2) \qquad k_1, k_2 \in \mathbb{Z} \tag{A.26}$$

The components of these lattice points are:

$$\boldsymbol{\omega}|_1 = \begin{pmatrix} k_1(\Delta_1^2\cos(\theta) - \Delta_2^2\sin(\theta)) + k_2(\Delta_1^2\cos(\theta) + \Delta_2^2\sin(\theta)) \\ k_1(\Delta_1^2\sin(\theta) + \Delta_2^2\cos(\theta)) + k_2(\Delta_1^2\sin(\theta) - \Delta_2^2\cos(\theta)) \end{pmatrix}, \qquad k_1, k_2 \in \mathbb{Z} \tag{A.27}$$

Notice that the 2 terms in the first component coincide with the left-hand-sides of (2.39) and (2.40) and the 2 terms in the second component coincide with the left-hand-sides of (2.41) and (2.42). Applying Lemma 3, if Q^2 is not constructed so that the EVQ is a periodic quantizer, that is, if $\boldsymbol{M}_{\Lambda^2} \neq diag[1/d_1^2, 1/d_2^2]\boldsymbol{M}_{S\Lambda^2}$, with $S\Lambda^2$ being a geometrically scaled-similar sublattice of Λ^1, at least one of the following equations is **not** satisfied:

$$\begin{aligned}
\text{(first component in } \boldsymbol{w}) \quad \Delta_1^2\cos(\theta) - \Delta_2^2\sin(\theta) &= q_1\Delta_1^1 && \text{(A.28)}\\
\text{(first component in } \boldsymbol{w}) \quad \Delta_1^2\cos(\theta) + \Delta_2^2\sin(\theta) &= q_2\Delta_1^1 && \text{(A.29)}\\
\text{(second component in } \boldsymbol{w}) \quad \Delta_1^2\sin(\theta) + \Delta_2^2\cos(\theta) &= q_3\Delta_2^1 && \text{(A.30)}\\
\text{(second component in } \boldsymbol{w}) \quad \Delta_1^2\sin(\theta) - \Delta_2^2\cos(\theta) &= q_4\Delta_2^1 && \text{(A.31)}\\
\text{where} \quad q_1, q_2, q_3, q_4 \in \mathcal{Q}
\end{aligned}$$

that is, at least one $q_i \notin \mathcal{Q}$. We now recall one of the properties of the *mod* function, which is that if $z = \mu v$ where $\mu \notin \mathcal{Q}$ and $v \in \mathbb{R}$, then $\{kz \; mod \; v, k \in \mathbb{Z}\} =]0, v[$. In the case of having $z = qv$ with $q \in \mathcal{Q}$, then, the set $\{kz \; mod \; v, k \in \mathbb{Z}\}$ is composed only of a finite number of distinct values. This gives 2 cases: Case 1: if at least one of the equations (2.39),(2.40) is not satisfied, then the first (horizontal) component in (A.27) of the lattice points of Λ^2 can have an arbitrary value (modulo Δ_1^1) (see Fig. A.2) and Case 2: if at least one of the equations (2.41),(2.42) is not satisfied, then the second (vertical) component in (A.27) can have an arbitrary value (modulo Δ_2^1). Notice that Case 1 and Case 2 are equivalent because the only difference between them is which coordinate fails to have a finite number of different values. Case 1 is the one that is actually represented graphically in Fig. A.2, and Case 2 corresponds to the Fig. A.2 rotated 90 degrees. Thus, the proof of Case 1 and Case 2 is exactly the same, and we can consider only Case 1 without loss of generality.

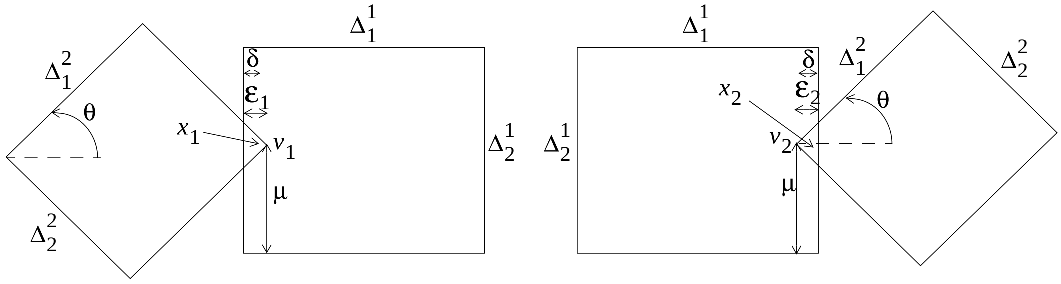


Figure A.2: Case 1 (Case 2 is Case 1 rotated 90 degrees) in the proof of the Theorem. It is not possible (if the EVQ is not periodic) to keep linear consistency simmultaneously in the 2 (small) EVQ cells shown.

Thus, consider that at least one of the equations (2.39),(2.40) is not satisfied and also let first both (2.41),(2.42) be satisfied, thus allowing a finite number of values (modulo Δ_2^1) in the second component.

Then, if we apply the previous property of the *mod* function, we can find a vertex $\boldsymbol{v}$ of Λ^2 of the form:

$$\boldsymbol{v}|_1 = \left(\begin{array}{c} I_x\Delta_1^1 + \gamma \\ y \end{array} \right), \forall \gamma \in]0, \Delta_1^1[, \text{ where } I_y\Delta_2^1 \leq y < (I_y+1)\Delta_2^1 \text{ , } I_x, I_y \in \mathbb{Z} \tag{A.32}$$

Consider now the following 2 input vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ defined as follows:

$$\begin{array}{l} \boldsymbol{x}_1|_1 = \boldsymbol{v}_1|_1 - \begin{pmatrix} \delta \\ 0 \end{pmatrix} = \begin{pmatrix} I_{x_1}\Delta_1^1 + \epsilon_1 - \delta \\ y_1 \end{pmatrix} \\ \boldsymbol{x}_2|_1 = \boldsymbol{v}_2|_1 + \begin{pmatrix} \delta \\ 0 \end{pmatrix} = \begin{pmatrix} (I_{x_2}+1)\Delta_1^1 - \epsilon_2 + \delta \\ y_2 \end{pmatrix} \end{array}$$

where

$$\boldsymbol{v}_1 = \boldsymbol{v}_{\{\gamma=\epsilon_1\}},\ \boldsymbol{v}_2 = \boldsymbol{v}_{\{\gamma=\Delta_1^1-\epsilon_2\}}, \Delta_1^1 \gg \epsilon_1, \epsilon_2 > 0, \Delta_1^1 \gg \delta > 0, \delta < min(\epsilon_1, \epsilon_2),$$
$$I_{y_1}\Delta_2^1 \le y_1 < (I_{y_1}+1)\Delta_2^1, \quad I_{y_2}\Delta_2^1 \le y_2 < (I_{y_2}+1)\Delta_2^1 \tag{A.33}$$

where $I_{x_1}, I_{x_2}, I_{y_1}, I_{y_2} \in \mathcal{Z}$.

If we apply the quantizers Q^1 and Q^2 to the input vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and then take the average, the final reconstructions $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ given by the EVQ are:

$$\hat{\boldsymbol{x}}_1|_1 = \begin{pmatrix} I_{x_1}\Delta_1^1 + \frac{\Delta_1^1}{4} - \frac{\Delta_1^2\cos(\theta)}{4} - \frac{\Delta_2^2\sin(\theta)}{4} + \frac{\epsilon_1}{2} \\ \frac{I_{y_1}\Delta_2^1}{2} + \frac{\Delta_2^1}{4} - \frac{\Delta_1^2\sin(\theta)}{4} + \frac{\Delta_2^2\cos(\theta)}{4} + \frac{y_1}{2} \end{pmatrix} \tag{A.34}$$

$$\hat{\boldsymbol{x}}_2|_1 = \begin{pmatrix} I_{x_2}\Delta_1^1 + \frac{3\Delta_1^1}{4} + \frac{\Delta_1^2\cos(\theta)}{4} + \frac{\Delta_2^2\sin(\theta)}{4} - \frac{\epsilon_2}{2} \\ \frac{I_{y_2}\Delta_2^1}{2} + \frac{\Delta_2^1}{4} + \frac{\Delta_1^2\sin(\theta)}{4} - \frac{\Delta_2^2\cos(\theta)}{4} + \frac{y_2}{2} \end{pmatrix} \tag{A.35}$$

In order to be able to express the constraints to satisfy consistency along the 2 directions determined by the second basis $\{\boldsymbol{\varphi}_1^2, \boldsymbol{\varphi}_2^2\}$, we also express $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ with their components given with respect to this second basis (this is actually equivalent to a clockwise rotation of the plane by an angle of θ):

$$\hat{\boldsymbol{x}}_1|_2 = \begin{pmatrix} I_{x_1}\Delta_1^1\cos(\theta) + \frac{I_{y_1}\Delta_2^1\sin(\theta)}{2} + \frac{\Delta_1^1\cos(\theta)}{4} + \frac{\Delta_2^1\sin(\theta)}{4} - \frac{\Delta_1^2}{4} + \frac{\sin(\theta)y_1}{2} + \frac{\cos(\theta)\epsilon_1}{2} \\ -I_{x_1}\Delta_1^1\sin(\theta) + \frac{I_{y_1}\Delta_2^1\cos(\theta)}{2} - \frac{\Delta_1^1\sin(\theta)}{4} + \frac{\Delta_2^1\cos(\theta)}{4} + \frac{\Delta_2^2}{4} + \frac{\cos(\theta)y_1}{2} - \frac{\sin(\theta)\epsilon_1}{2} \end{pmatrix}$$

$$\hat{\boldsymbol{x}}_2|_2 = \begin{pmatrix} I_{x_2}\Delta_1^1\cos(\theta) + \frac{I_{y_2}\Delta_2^1\sin(\theta)}{2} + \frac{3\Delta_1^1\cos(\theta)}{4} + \frac{\Delta_2^1\sin(\theta)}{4} + \frac{\Delta_1^2}{4} + \frac{\sin(\theta)y_2}{2} - \frac{\cos(\theta)\epsilon_2}{2} \\ -I_{x_2}\Delta_1^1\sin(\theta) + \frac{I_{y_2}\Delta_2^1\cos(\theta)}{2} - \frac{3\Delta_1^1\sin(\theta)}{4} + \frac{\Delta_2^1\cos(\theta)}{4} - \frac{\Delta_2^2}{4} + \frac{\cos(\theta)y_2}{2} + \frac{\sin(\theta)\epsilon_2}{2} \end{pmatrix}$$

For notational convenience, assume that the symbols $\le, <$ are component-wise relation symbols. Then, all the constraints that have to be satisfied to achieve consistency are given by the following component-wise inequalities:

$$\begin{pmatrix} I_{x_1}\Delta_1^1 \\ I_{y_1}\Delta_2^1 \end{pmatrix} \le \quad \hat{\boldsymbol{x}}_1|_1 \quad < \begin{pmatrix} I_{x_1}\Delta_1^1 + \Delta_1^1 \\ I_{y_1}\Delta_2^1 + \Delta_2^1 \end{pmatrix} \tag{A.36}$$

$$\begin{pmatrix} I_{x_2}\Delta_1^1 \\ I_{y_2}\Delta_2^1 \end{pmatrix} \le \quad \hat{\boldsymbol{x}}_2|_1 \quad < \begin{pmatrix} I_{x_2}\Delta_1^1 + \Delta_1^1 \\ I_{y_2}\Delta_2^1 + \Delta_2^1 \end{pmatrix} \tag{A.37}$$

$$\boldsymbol{v}_1|_2 - \begin{pmatrix} \Delta_1^2 \\ 0 \end{pmatrix} \le \quad \hat{\boldsymbol{x}}_1|_2 \quad < \boldsymbol{v}_1|_2 + \begin{pmatrix} 0 \\ \Delta_2^2 \end{pmatrix} \tag{A.38}$$

$$\boldsymbol{v}_2|_2 - \begin{pmatrix} 0 \\ \Delta_2^2 \end{pmatrix} \leq \quad \hat{\boldsymbol{x}}_2|_2 \quad < \boldsymbol{v}_2|_2 + \begin{pmatrix} \Delta_1^2 \\ 0 \end{pmatrix} \tag{A.39}$$

where

$$\begin{aligned} \boldsymbol{v}_1|_2 &= \begin{pmatrix} \cos(\theta) I_{x_1}\Delta_1^1 + \sin(\theta) y_1 + \cos(\theta)\epsilon_1 \\ -\sin(\theta) I_{x_1}\Delta_1^1 + \cos(\theta) y_1 - \sin(\theta)\epsilon_1 \end{pmatrix}, \\ \boldsymbol{v}_2|_2 &= \begin{pmatrix} \cos(\theta)(I_{x_2}+1)\Delta_1^1 + \sin(\theta) y_2 - \cos(\theta)\epsilon_2 \\ -\sin(\theta)(I_{x_2}+1)\Delta_1^1 + \cos(\theta) y_2 + \sin(\theta)\epsilon_2 \end{pmatrix} \end{aligned}$$

From the first component inequality in either (A.36) or (A.37), and using the fact that $\epsilon_1 > 0$ and $\epsilon_2 > 0$ can be taken as small as we want, we get the following lower bound for Δ_1^1:

$$\Delta_1^1 \geq \Delta_1^2 \cos(\theta) + \Delta_2^2 \sin(\theta) \tag{A.40}$$

Similarly, from (A.38) and (A.39), we can obtain, after operating, lower bounds for Δ_1^2 and Δ_2^2. Let $\mu_i = y_i \ mod \ \Delta_2^1$, $i = 1, 2$, that is, $\mu_1 = y_1 - I_{y_1}\Delta_2^1$ and $\mu_2 = y_2 - I_{y_2}\Delta_2^1$. The actual lower bounds for Δ_1^2 and Δ_2^2 depend on the parameters μ_1 and μ_2:

$$\Delta_1^2 \geq \Delta_1^1 \cos(\theta) + \Delta_2^1 \sin(\theta) - 2\sin(\theta)\mu_1 \quad (\text{tightest if } \mu_1 \leq \frac{\Delta_2^1}{2}) \tag{A.41}$$

$$\Delta_2^2 \geq \Delta_1^1 \sin(\theta) + \Delta_2^1 \cos(\theta) - 2\cos(\theta)\mu_2 \quad (\text{tightest if } \mu_2 \leq \frac{\Delta_2^1}{2}) \tag{A.42}$$

$$\Delta_1^2 \geq \Delta_1^1 \cos(\theta) - \Delta_2^1 \sin(\theta) + 2\sin(\theta)\mu_2 \quad (\text{tightest if } \mu_2 \geq \frac{\Delta_2^1}{2}) \tag{A.43}$$

$$\Delta_2^2 \geq \Delta_1^1 \sin(\theta) - \Delta_2^1 \cos(\theta) + 2\cos(\theta)\mu_1 \quad (\text{tightest if } \mu_1 \geq \frac{\Delta_2^1}{2}) \tag{A.44}$$

We show next that it is always possible to find points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ such that $\mu_1 = \mu_2$. Since (A.30) and (A.31) are satisfied, let $q_3 = \frac{n_1}{m_1}$ and $q_4 = \frac{n_2}{m_2}$, such that $gcd(n_1, m_1) = 1$ and $gcd(n_2, m_2) = 1$. Then, we have that:

$$\begin{aligned} &\{k_1(\Delta_1^2 \sin(\theta) + \Delta_2^2 \cos(\theta)) + k_2(\Delta_1^2 \sin(\theta) - \Delta_2^2 \cos(\theta)) \quad mod \quad \Delta_2^1\} = \\ &\left\{0, \frac{\Delta_2^1}{m_1 m_2}, \frac{2\Delta_2^1}{m_1 m_2}, \cdots, \frac{(m_1 m_2 - 1)\Delta_2^1}{m_1 m_2}\right\} \end{aligned} \tag{A.45}$$

This directly implies that we can always (by varying k_1 and k_2 in (A.45)) find 2 vertices $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ (satisfying that $\mu_1 = \mu_2 = \mu$) of the following form:

$$\boldsymbol{v}_1|_1 = \begin{pmatrix} I_{x_1}\Delta_1^1 + \epsilon_1 \\ I_{y_1}\Delta_2^1 + \mu \end{pmatrix} \qquad \boldsymbol{v}_2|_1 = \begin{pmatrix} (I_{x_2}+1)\Delta_1^1 - \epsilon_2 \\ I_{y_2}\Delta_2^1 + \mu \end{pmatrix} \tag{A.46}$$

Consider now the case that at least one of the equations (A.30), (A.31) were not satisfied. Then, it is also clear that we could find cells with values of μ_1 and μ_2 as close to each other as wanted because we have a continuum of values (modulo Δ_2^1) in this component, and the same conclusions in the proof would follow.

Consider first the case of $\mu_1 = \mu_2 = \mu \leq \frac{\Delta_2^1}{2}$. In this case, if we multiply (A.41) and (A.42) by $\cos(\theta)$ and $\sin(\theta)$ respectively and then we sum them, making use of the equality $\cos^2(\theta) + \sin^2(\theta) = 1$, we obtain an upper bound for Δ_1^1 given by:

$$\Delta_1^1 \leq \Delta_1^2 \cos(\theta) + \Delta_2^2 \cos(\theta) + \sin(2\theta)(2\mu - \Delta_2^1) \tag{A.47}$$

In order for the upper (A.47) and lower (A.40) bounds of Δ_1^1 to be consistent[1], it is necessary to have $\mu \geq \frac{\Delta_2^1}{2}$, which implies that the only valid value for μ is $\mu = \frac{\Delta_2^1}{2}$. Consider now the case of $\mu_1 = \mu_2 = \mu \geq \frac{\Delta_2^1}{2}$. In the same way, if we multiply (A.43) and (A.44) by $\cos(\theta)$ and $\sin(\theta)$ respectively and we sum them, we obtain an upper bound for Δ_1^1 given by:

$$\Delta_1^1 \leq \Delta_1^2 \cos(\theta) + \Delta_2^2 \sin(\theta) + \sin(2\theta)(\Delta_2^1 - 2\mu) \tag{A.48}$$

As before, for the upper (A.48) and lower (A.41) bounds to be consistent, we need $\mu \leq \frac{\Delta_2^1}{2}$, which implies again that $\mu = \frac{\Delta_2^1}{2}$.

Thus, in order to achieve consistency simultaneously for the input vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, as defined in (A.46), it is necessary to have always $\mu = \frac{\Delta_2^1}{2}$. But this is clearly impossible because, for instance, by taking vertices with k_1,k_2 given by $k_1 = l_1 m_1$ and $k_2 = l_2 m_2$, $l_1, l_2 \in \mathbb{Z}$ in (A.45), we have always $\mu = 0$. Therefore, we conclude that it is impossible to achieve consistency for the 2 input vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ simultaneously ∎.

A.8 Basis for the sum of 2 lattices: gcld method

Let $\boldsymbol{M}_{\Lambda^1}$ and $\boldsymbol{M}_{\Lambda^2}$ be the generator matrices of Λ^1 and Λ^2 respectively. If $\Lambda^\Sigma = \Lambda^1 + \Lambda^2$ is a lattice, then $\boldsymbol{M}_{\Lambda^\Sigma} = gcld(\boldsymbol{M}_{\Lambda^1}, \boldsymbol{M}_{\Lambda^2})$ [47]. We need to introduce first a few concepts. Given matrices $\boldsymbol{A}, \boldsymbol{D} \in \mathbb{R}$, $\boldsymbol{D}$ is a left divisor of $\boldsymbol{A}$ and $\boldsymbol{A}$ is a right multiple of $\boldsymbol{D}$, if there exists a matrix $\boldsymbol{C} \in \mathbb{Z}^{N\times N}$ such that $\boldsymbol{A} = \boldsymbol{DC}$. If Λ^Σ exists, which is equivalent to having that $(\boldsymbol{M}_{\Lambda^1})^{-1}\boldsymbol{M}_{\Lambda^2} \in \mathbb{Q}^{N\times N}$, then a matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a common left divisor of $\boldsymbol{M}_{\Lambda^1}$ and $\boldsymbol{M}_{\Lambda^2}$ if $\boldsymbol{M}_{\Lambda^1} = \boldsymbol{DP}$ and $\boldsymbol{M}_{\Lambda^2} = \boldsymbol{DQ}$, where $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{Z}^{N\times N}$. A matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a greatest common left divisor (gcld) of $\boldsymbol{M}_{\Lambda^1}$ and $\boldsymbol{M}_{\Lambda^2}$ if a) it is a common left divisor of $\boldsymbol{M}_{\Lambda^1}$ and

[1] Notice that $\theta \in]0, \frac{\pi}{2}[$, which means that $\sin(2\theta) > 0$.

$\boldsymbol{M}_{\Lambda^2}$ and b) $\boldsymbol{D}$ is a right multiple of every common left divisor of $\boldsymbol{A}$ and $\boldsymbol{B}$. Two matrices $\boldsymbol{H}, \boldsymbol{K} \in \mathbb{Z}^{N\times N}$ are left coprime if every gcld of $\boldsymbol{H}$ and $\boldsymbol{K}$ is unimodular.

Given $\boldsymbol{E}$, any ordered pair of left coprime matrices $(\boldsymbol{H}, \boldsymbol{K})$ in $\mathbb{Z}^{N\times N}$ with $det(\boldsymbol{H} \neq 0$, such that $\boldsymbol{E} = \boldsymbol{H}^{-1}\boldsymbol{K}$, is called a left coprime factorization of $\boldsymbol{E}$. It is easy to see that given matrices $\boldsymbol{M}_{\Lambda^1}, \boldsymbol{M}_{\Lambda^2} \in \mathbb{R}^{N\times N}$, with $(\boldsymbol{M}_{\Lambda^1})^{-1}\boldsymbol{M}_{\Lambda^2} \in \mathbb{Q}^{N\times N}$, if $\boldsymbol{D} = gcld(\boldsymbol{M}_{\Lambda^1}, \boldsymbol{M}_{\Lambda^2})$, it is $\boldsymbol{D} = \boldsymbol{A}\boldsymbol{H}^{-1} = \boldsymbol{B}\boldsymbol{K}^{-1}$, where $(\boldsymbol{H}, \boldsymbol{K})$ is a left coprime factorization of $\boldsymbol{E} = (\boldsymbol{M}_{\Lambda^1})^{-1}\boldsymbol{M}_{\Lambda^2}$.

In order to find the left coprime factorization of a full rank matrix $\boldsymbol{E} \in \mathbb{Q}^{N\times N}$, one can use the following algorithm [119]. Let $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{Z}^{N\times N}$ be unimodular matrices such that $\boldsymbol{UEV}$ is in Smith-McMillan normal form, that is:

$$\boldsymbol{UEV} = (\boldsymbol{S}_b)^{-1}\boldsymbol{S}_a\,, \quad \boldsymbol{S}_b = \mathrm{diag}[b_1, \cdots, b_N]\,, \boldsymbol{S}_a = \mathrm{diag}[a_1, \cdots, a_N] \tag{A.49}$$

where $a_i, b_i \in \mathbb{Z}$ are coprime, $a_{i+1}|a_i$, $b_i|b_{i+1}$, $i = 1, \cdots, N$. Then, taking $\boldsymbol{H} = \boldsymbol{S}_b\boldsymbol{U}$ and $\boldsymbol{K} = \boldsymbol{S}_a\boldsymbol{V}^{-1}$, $(\boldsymbol{H}, \boldsymbol{K})$ is a left coprime factorization of $\boldsymbol{E}$.

Appendix B

B.1 Proof of Proposition 1

The hypotheses guarantee that $(\Lambda^j)^* \cong \Lambda^j \cong K^{n/\kappa}$, and, by the Theorem 3, $\Lambda = \Lambda^1 \cap \cdots \cap \Lambda^r$, where r = number of minimal vectors of Λ^* divided by the number of minimal vectors of $(K)^{N/\kappa}$ ■.

B.2 Proof of Theorem 4

Let p_i $(i = 1, \ldots, k)$ be the probability that a randomly chosen point in $\mathbb{R}^n$ (uniformly distributed over a very large ball, say) belongs to $\mathcal{P}_i$, and let $n_i = p_i V/V_i$, where V_i is the volume of $\mathcal{P}_i$ and $V = det(M_\Lambda)$ is the volume of a fundamental region or Voronoi cell for Λ. Let $\mathcal{V}$ be the particular Voronoi cell for Λ that contains the origin. Then the periodic tesselation is periodic with "tile" equal to the part lying in $\mathcal{V}$, and there are n_1 cells of type $\mathcal{P}_1$ per copy of $\mathcal{V}$, n_2 cells of type $\mathcal{P}_2$, etc. Also

$$V = n_1 V_1 + \cdots + n_k V_k \ . \tag{B.1}$$

We assume that a point that falls into a cell of type $\mathcal{P}_i$ is quantized as the centroid $\boldsymbol{c}_i$ of that cell, in which case the mean squared error is

$$U_i = \int_{\mathcal{P}_i} \|\boldsymbol{x} - \boldsymbol{c}_i\|^2 d\boldsymbol{x} \ .$$

The mean squared error for the full quantizer is then

$$U = \sum_{i=1}^{k} n_i U_i = V \sum_{i=1}^{k} \frac{p_i U_i}{V_i} \ . \tag{B.2}$$

We now derive the expression that we will use as a measure of the normalized mean squared error of this quantizer. Suppose we are quantizing a random variable $X \in \mathbb{R}^N$, with differential entropy per dimension $h(X)$, and whose support contains a large number of points of Λ.

A general theorem of Zador about vector quantizers ([135], [136], [62, Eq. (20)]) implies that in this "high-rate" case the average mean squared error per dimension $U/(NV)$ can be approximated by

$$\frac{U}{NV} \approx G\, 2^{2(h(X)-R)} \ , \tag{B.3}$$

where R bits/symbol is the quantizing rate and G depends on the positions of the quantizing points but is independent of X.

Since G does not depend on the distribution of X, we may choose any convenient distribution in order to calculate G, and we assume that X is uniformly distributed over a large region of $\mathbb{R}^N$, or, equivalently, that X is uniformly distributed over V. Then

$$h(X) = \frac{1}{N} \log_2 V, \quad \text{so} \quad 2^{2h(X)} = V^{2/N} \ . \tag{B.4}$$

To calculate R, observe that we need $h(p_1, \ldots, p_k) = -\sum_{i=1}^k p_i \log_2 p_i$ bits to specify the type of cell to which the quantized point belongs, and a further $\sum_{i=1}^k p_i \log_2 n_i = \sum_{i=1}^k p_i \log_2 (p_i V/V_i)$ bits to specify the particular one of the n_i cells of that type. This requires a total of $\log_2 V - \sum_{i=1}^k p_i \log_2 V_i$ bits, and then R is this quantity divided by N, so that

$$2^{-2R} = V^{-2/N} \prod_{k=1}^{k} V_i^{2p_i/N} \ . \tag{B.5}$$

From (B.2)–(B.5) we obtain

$$G = \frac{\sum_{i=1}^k \frac{p_i U_i}{V_i}}{N \left(\prod_{i=1}^k V_i^{p_i} \right)^{\frac{2}{N}}} \ , \tag{B.6}$$

the normalized mean squared error per dimension, which we take as our figure of merit for a quantizer. The numerator of (B.6) is equal to U/V (see (B.2)) ▪.

Appendix C

C.1 Proof of Lemma 5

The set $S_2 = Q^{-1}(\boldsymbol{c}_q(\boldsymbol{x}_o))$ is clearly convex because it is a K-dimensional cube in $\mathbb{R}^K$. In other words, it is a closed linear manifold in the Hilbert space $\mathbb{R}^K$, which ensures convexity. Similarly, the set $W_2 = \mathit{span}\{\alpha_1(\phi), \cdots, \alpha_J(\phi)\}$ is a closed linear manifold in the Hilber space $L^2(\mathbb{R})$, which also implies convexity $\blacksquare$.

C.2 Proof of Theorem 6

The solution of the linear programming problem is very simple. Consider, for each ϕ, the quantity $Z = \sum_{i=1}^{J} \beta_i \alpha_i(\phi)$ where β_i can take only one of 2 values, namely, l_i or u_i, $i = 1, \cdots, J$, where $\Delta(\phi) = u_i - l_i$. For each ϕ, $R_U(\boldsymbol{x}_o, \phi)$ is obtained choosing for β_i the bounds of the quantization intervals $[l_i\, u_i]$, $i = 1, \cdots, J$ which makes Z have the largest posible value (towards $+\infty$). For each ϕ, $R_L(\boldsymbol{x}_o, \phi)$ is obtained choosing for β_i the bounds of the quantization intervals $[l_i\, u_i]$, $i = 1, \cdots, J$ which makes Z have the smallest posible value (towards $-\infty$). Thus, if it happens that $\alpha_i(\phi) > 0$, then, $R_U(\boldsymbol{x}_o, \phi)$ is obtained taking $\beta_i = u_i$ and $R_L(\boldsymbol{x}_o, \phi)$ is obtained taking $\beta_i = l_i$. If $\alpha_i(\phi) < 0$, the mapping is reversed.

From the solution of the linear programming problem, it is clear that

$$\mathrm{Width}(\boldsymbol{x}_o, \phi) = R_U(\boldsymbol{x}_o, \phi) - R_L(\boldsymbol{x}_o, \phi) = \sum_{i=1}^{J} \Delta(\phi_i)|\alpha_i(\phi)|$$

because, by linearity, for a given ϕ, the i-term in $\mathrm{Width}(\boldsymbol{x}_o, \phi)$ will be either $(u_i - l_i)\alpha_i(\phi)$ if $\alpha_i(\phi) > 0$, or $(l_i - u_i)\alpha_i(\phi)$ if $\alpha_i(\phi) < 0$. In both cases, we obtain $(u_i - l_i)|\alpha_i(\phi)| = \Delta(\phi_i)|\alpha_i(\phi)|$ and the result follows.

The central curve $\hat{c}(\boldsymbol{x}_o, \phi)$ is given by $\hat{c}(\boldsymbol{x}_o, \phi) = R_L(\boldsymbol{x}_o, \phi) + \frac{\text{Width}(\boldsymbol{x}_o, \phi)}{2}$, which implies that:

$$\hat{c}(\boldsymbol{x}_o, \phi) = \sum_{i=1}^{J} \beta_i \alpha_i(\phi) \quad \text{where} \quad \beta_i = \left\{ \begin{array}{ll} \left(l_i + \frac{\Delta(\phi_i)}{2}\right) = \left(\frac{l_i + u_i}{2}\right) & \text{if } \alpha_i(\phi) > 0 \\ \left(u_i - \frac{\Delta(\phi_i)}{2}\right) = \left(\frac{l_i + u_i}{2}\right) & \text{if } \alpha_i(\phi) < 0 \end{array} \right\}$$

Finally, since $\hat{c}(\boldsymbol{x}_o, \phi)$ is the central curve in $R(\boldsymbol{x}_o, \phi)$, we have that $R_U(\boldsymbol{x}_o, \phi) = \hat{c}(\boldsymbol{x}_o, \phi) + \frac{\text{Width}(\boldsymbol{x}_o, \phi)}{2}$ and also that $R_L(\boldsymbol{x}_o, \phi) = \hat{c}(\boldsymbol{x}_o, \phi) - \frac{\text{Width}(\boldsymbol{x}_o, \phi)}{2}$. The result follows ■.

C.3 Proof of Fact 3

The proof follows by simple computation. The transform coefficient value $c^l(\boldsymbol{x}_o, \phi)$ for location $\boldsymbol{x}_o$, angle ϕ and level l, is given by $c^l(\boldsymbol{x}_o, \phi) = \sum_{i=1}^{J} \alpha_i(\phi) c^l(\boldsymbol{x}_o, \phi_i)$, where $\phi_1, \cdots, \phi_J$ are the basic angles. Notice that:

$$(c^l(\boldsymbol{x}_o, \phi))^2 = \left(\sum_{i=1}^{J} \alpha_i(\phi) c^l(\boldsymbol{x}_o, \phi_i) \right)^2 = \sum_{i=1} \sum_{k=1} \alpha_i(\phi) \alpha_k(\phi) c^l(\boldsymbol{x}_o, \phi_i) c^l(\boldsymbol{x}_o, \phi_k) \tag{C.1}$$

This is a quadratic form with a symmetric matrix $\boldsymbol{C}^l$ with components being $C^l_{ik} = c^l(\boldsymbol{x}_o, \phi_i) c^l(\boldsymbol{x}_o, \phi_k)$. Averaging over all the spatial locations in level l, the result follows ■.

C.4 Proof of Proposition 2

Let $\phi_1, \cdots, \phi_J$ be the basic angles, respect to which both correlation matrices $\boldsymbol{C}^l_I$ and $\boldsymbol{C}^l_{I_\theta}$ are calculated. We denote by $c_\theta(\boldsymbol{x}_o, \phi)$ the transform coefficient value for the rotated texture I_θ at a location $\boldsymbol{x}_o$ and absolute angle ϕ. Notice that a counter-clockwise rotation of I (in order to get I_θ) is equivalent to leaving fixed I and rotating clockwise 2D axes by an angle $-\theta$, that is:

$$c_\theta(\boldsymbol{x}_o, \phi_k) = c^l(\boldsymbol{x}_o, \phi_k - \theta) = \sum_{i=1}^{J} \alpha_i(\phi_k - \theta) c^l(\boldsymbol{x}_o, \phi_i), \quad i = 1, \cdots, J \tag{C.2}$$

This implies that:

$$c_\theta(\boldsymbol{x}_o, \phi_n) c_\theta(\boldsymbol{x}_o, \phi_m) = \sum_{i=1}^{J} \sum_{k=1}^{J} \alpha_i(\phi_n - \theta) \alpha_k(\phi_m - \theta) c^l(\boldsymbol{x}_o, \phi_i) c^l(\boldsymbol{x}_o, \phi_k) \tag{C.3}$$

From here, it follows directly that $\boldsymbol{C}^l_{I_\theta} = \boldsymbol{R}(\theta)\boldsymbol{C}^l_I\boldsymbol{R}^T(\theta)$, where $\boldsymbol{R}(\theta)$ is given by:

$$\boldsymbol{R}(\theta) = \begin{pmatrix} \alpha_1(\phi_1 - \theta) & \alpha_2(\phi_1 - \theta) & \cdots & \alpha_J(\phi_1 - \theta) \\ \alpha_1(\phi_2 - \theta) & \alpha_2(\phi_2 - \theta) & \cdots & \alpha_J(\phi_2 - \theta) \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_J(\phi_J - \theta) & \cdots & \cdots & \alpha_J(\phi_J - \theta) \end{pmatrix}$$

When the J basic angles are equispaced, using (4.16), it can be trivially checked that $\boldsymbol{R}(\theta)$ is an orthogonal matrix ▪.