CONTEXT QUANTIZATION FOR ADAPTIVE ENTROPY CODING IN IMAGE COMPRESSION

by

Tong Jin
B.A.Sc., Tianjin University,1995
M.A.Sc., Chinese Academy of Sciences, 1998

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

In the School
of
Engineering Science

© Tong Jin 2006

SIMON FRASER UNIVERSITY

Fall 2006

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without permission of the author.

APPROVAL

Name: **Tong Jin**

Degree: **Doctor of Philosophy**

Title of Thesis: **Context Quantization for Adaptive Entropy Coding in Image Compression**

Examining Committee:

Chair: **Dr. Dong In Kim**

____

Dr. Jie Liang, Senior Supervisor, SFU

____

Dr. Xiaolin Wu, Co-Supervisor, McMaster University

____

Dr. Richard (Hao) Zhang, Supervisor

____

Dr. Ze-Nian Li, Supervisor

____

Dr. Ivan Bajic, Internal Examiner

____

Dr. Lina J. Karam, External Examiner, Arizona State University

Date Approved: Dec. 12, 2006

SIMON FRASER UNIVERSITY library

DECLARATION OF PARTIAL COPYRIGHT LICENCE

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Revised: Fall 2006

ABSTRACT

Context based adaptive entropy coders are used in newer compression standards to achieve rates that are asymptotically close to the source entropy: separate arithmetic coders are used for a large number of possible conditioning classes. This greatly reduces the amount of sample data available for learning. To combat this problem, which is referred as the context dilution problem in the literature, one needs to balance the benefit of using high-order context modeling and the learning cost associated with context dilution.

In the first part of this dissertation, we propose a context quantization method to attack the context dilution problem for non-binary source. It begins with a large number of conditioning classes and then uses a clustering procedure to reduce the number of contexts into a desired size. The main operational difficulty in practice is how to describe the complex partition of the context space. To deal with this problem, we present two novel methods, coarse context quantization (CCQ) and entropy coded state sequence (ECSS), for efficiently describing the context book, which completely specifies the context quantizer mappings information.

The second part of this dissertation considers binarization of non-binary sources. Same as non- binary source, the cost of sending the complex context description as side information is very high. Up to now, all the context quantizers are designed off-line and being optimized with respect to the statistics of the training set. The problem of handling the mismatch between the training set and an input image has remained largely untreated. We propose three novel schemes, minimum description length, image dependent and minimum adaptive code length, to deal with this problem. The experimental results show that our approach outperforms the JBIG and JBIG2 standard with peak compression improvement of 24% and 11% separately on the chosen set of halftone images.

In the third part of this dissertation, we extend our study to the joint design of both quantizers and entropy coders. We propose a context-based classification and adaptive quantization scheme, which essentially produce a finite state quantizer and entropy coder with the same procedure.

Keywords: context, entropy coding, context quantization, image compression.

DEDICATION

To the memory of Dr. Jacques Vaisey

ACKNOWLEDGMENT

Prayerful thanks to our merciful GOD who give me everything I have, held my hands and led me through the darkest time in my life.

My foremost thanks go to my former senior supervisor Dr. Jacques Vaisey who, unfortunately, passed away three years before my defence. He was a great professor, a great teacher and, above all, a great man. I thank him for introducing me into the world of compression and invaluable guidance.

I am sincerely grateful to my senior supervisor Dr. Jie Liang, for his supervision, kind advice, patience and encouragement that carried me on through difficult times.

I would like to express my deep gratitude to my co-supervisor Dr. Xiaolin Wu for his insights and suggestions that helped to shape my research skills. Without him, this dissertation would not have been possible. His valuable feedback contributed greatly to this dissertation.

Special thanks to my supervisory committee members, Dr. Richard Zhang and Dr. Ze-Nian Li for their comments on my dissertation. Also thanks to Dr. Lina J. Karam and Dr. Ivan Bajic for taking time to act as thesis examiner and to Dr. Dong In Kim for chairing the examine committee.

Grateful thanks also go to my past and present labmates and friends, Subbalakshmi, Florina Rogers, Jerry Zhang, Echo Ji, Ye Lu, Ed Chiu, Yi Zheng, Jamshid Ameli, Guoqian Sun and Upul Samarawicrama for always being there with help and encouragement.

My sincere appreciation goes to the brothers and sisters from SFU Chinese Christian Fellowship, Hui Qu and Fang Liu, Xiaofeng Zhang and Hui Zhang, Weitian Chen and Xiaoxiu Shi, Qingguo Li and Xuan Geng, Yifeng Huang and Jinyun Ren, Lingyun Ye, Yiduo Mao, and Yongmei Gong for walking with me, otherwise, I would have been lonely. Their loving friendship is priceless to me.

Finally, my heartfelt gratitude goes to my fiancé and my family. To my fiancé, Aldo Zeng, for the unconditional love he has given to me. I am indebted to my mom and dad, sister and brother-in-law for their supports, patience, encouragements and sacrifices over the years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1.
INTRODUCTION

1.1. Introduction

The idea of data compression is much older than our era of digital communications and computers. Since the early days of civilization, men have always been interested in economical communications. The purpose of data compression is to represent information using the minimum amount of medium so that it takes less time to transmit and less space to store. In ancient Greece, the cost of the papyrus or marble was far more expensive than today's paper. As a result, texts were written with no punctuation to save space. The ancient Chinese language was much more "compressed", but a bit harder to communicate in daily life, than its modern descendant. At that time concise expressions were necessary because the words could be only written on narrow bamboo plate. Abbreviations and acronyms have been used as a data compression technique for ages.

Data compression becomes more important in modern society due to the revolution of the information technology, which has changed the way we communicate. The emergence and development of the internet and the growing number of mobile phones and digital TV users are part of this revolution. Data compression has definitely had a very important role in the development of the multimedia technologies. In fact, without the current data compression techniques, the internet would not have the size and shape as it does today, and the mobile phones and digital TV's would not be as widespread as they are today. The music stored in CD's, the movies stored in DVD's and the images stored in digital cameras are all compressed.

A picture is worth one thousand words. An effective and popular form of modern digital communications is pictorial. One can hardly find a page in the internet that does not contain at least one image. Also, if we could search the information in the computers around the world, maybe it would be difficult to find a computer without image files stored in it.

Digital image compression, or digital image coding, is far more important than text compression because a digital image involves a large volume of data if uncompressed. Image compression has been an active research topic for more than 80 years, ever since the digitized pictures were first transmitted in the early 1920s.

Image compression techniques are also used in most of the video compression algorithms and standards. In fact, in many video compression standards, an image compression technique is used to code some non-consecutive video frames and the frames between them are interpolated using motion compensation techniques [1] to exploit the dependency between the frames.

Entropy coding is an important component in image and video coding systems. It performs lossless compression on symbols generated by a transform and quantization process to obtain a more efficient representation of source data. In the case where the source data possesses certain statistical dependencies between symbols, it is advantageous for an entropy coder to consider this statistical property in the source data. Furthermore, if an entropy coder can dynamically adapt to the statistics of the input symbols, better performance can be achieved than with its non-adaptive counterpart. Therefore, context-based adaptive entropy coding becomes an essential feature of modern image and video compression algorithms.

The design objective of context-based adaptive entropy coding is to asymptotically approach the source entropy. However, the adaptation takes time to converge to the source statistics and the compression performance suffers when the length of input data is relatively short. In image and video coding the problem is compounded by the fact that input sources contain significant memory (even after a decorrelation transform). A high-order context needs to be employed by conditional entropy coding to approach the source entropy: separate arithmetic coders are used for each of a large number of possible conditioning states (instances of a chosen context). This greatly reduces the amount of sample data available for learning. To deal with this problem, which is referred to as context dilution problem in the literature, one needs to balance the benefit of using high-order context modelling to fit the input data and the learning cost due to data dilution.

1.2. Main Contributions

In this thesis, first we attack the context dilution problem. Most of the former approaches define the context in an ad-hoc manner. In [2], we propose a context quantization method which begins with a large number of conditioning classes and then uses a clustering procedure to reduce the number of contexts to a desired value. We also show that the resulting context quantizer is optimal in the sense of minimizing the conditional entropy.

However, the main operational difficulty in practice, is how to describe the complex partition of the context space by the minimum conditional entropy context quantizer. The context-based adaptive entropy coder relies on this partition that maps the context space into coding states, which we call quantizer mapping. Two novel schemes are proposed to deal with the problem of quantizer mapping in [3]. Coarse context quantizer method is to decrease the size of the context space. Entropy coded state sequence method is designed for reducing the bits spending on individual entry of the context book. Some encouraging results are obtained.

We consider binarization of non-binary sources. Since the probability simplex space of a binary source is one dimensional, context quantizer design for binary sources is reduced to a scalar quantizer design problem. As a result, globally optimal context quantizers can be computed by dynamic programming. But, same as non-binary sources, the cost of transmitting inverse quantization in context space is very high. Therefore, up to now, all the context quantizers are designed off-line and being optimized with respect to the statistics of the training set. An ensuing question is how to handle any mismatch in statistics between the training set and an input image. This problem has remained largely untreated. To deal with it, we proposed three novel schemes in [4] and [5]. Minimum description length method is to minimize the sum of the bits emitted by the conditional entropy coder using the context quantizer and the side information to describe the context quantizer mappings. Image dependent context quantizer is designed based on input statistics alone to minimize the conditional entropy with small side information. Minimum adaptive code length context quantizer is aiming to minimize the effect of the mismatch between the training set and the input. The actual adaptive code length difference between the two sets, the training set plus the input and the training set alone, is minimized. Our experiments show that our schemes outperform JBIG and JBIG2 standard with peak compression improvement of 24% and 8% on the chosen set of twelve halftone images, which are among the most difficult binary sources to compress.

We also extend our work to the joint design of both quantizers and entropy coders. A context-based classification and adaptive quantization scheme on coefficient basis with non-parametric modelling is presented in [6] , which essentially produces a finite state quantizer and entropy coder with the same procedure. The results show that it has a great potential to improve the overall compression system performance.

1.3. Thesis Outline

Chapter 2 covers the fundamentals of context based entropy coding and quantization, which may be needed for understanding the research materials of the subsequent chapters.

Chapter 3 studies context quantizer design with the objective of optimizing context-based entropy coders for non-binary sources. An iterative algorithm to overcome the difficulty of context dilution is proposed. Furthermore, two novel schemes are developed for compactly describing the partition of the context space.

Chapter 4 is concerned with context quantizer design for binary sources. A new MDL (minimum description length) based adaptive context quantization scheme is presented first. An image dependent context quantizer with a very efficient m of side information is then described. Finally a context quantization method to deal with the mismatch statistics between the training set and the input image is proposed. This novel method optimizes the context quantizer under the criterion of minimum actual adaptive code length.

Chapter 5 studies the problem of joint design of both quantizer and entropy coder using context-based classification techniques. A novel non-parametric approach based on histogram quantization is proposed.

CHAPTER 2.
BACKGROUND REVIEW

2.1. Entropy Coding

2.1.1. Entropy

Entropy, a notion first introduced by Shannon [7] , is a measure of information. There is high amount of information in an event if the probability of the occurring of the event is low and vice versa. As an example, suppose you receive a phone call in January from a friend of yours in the Northern Territories. He says the weather there is very cold. This sentence really does not carry much information, as at that time of the year you do expect the weather to be cold there. However, if a top seed tennis player is beaten by a Wimbledon wildcard in the first round of tournament, all the sports channels will talk about the unexpected news, as there is high information in it.

Consider a memoryless source modelled by a discrete random variable X, with a symbol alphabet of size N, $\{x_0, x_1, ..., x_{N-1}\}$. The random variable X is characterized by its probability mass function

$$p_X(x_i) = prob(X = x_i) \qquad i = 0,1,\ldots,N-1 \tag{2.1}$$

The information content of the source is measured by its entropy (in the memoryless case the zeroth-order entropy) [8]

$$H(X) = -\sum_i p_X(x_i)\log_b p_X(x_i) \tag{2.2}$$

For binary systems, the logarithm base is two, and the unit of the entropy is bit.

2.1.2. Entropy Coding

Shannon [7] showed that the average number of bits necessary to encode a memoryless source without loss cannot be lower than the entropy of the source. However, the zeroth-order entropy does not take the memory among the source symbols into account. When memory is present in a source, dependencies between the symbols need to be exploited to achieve maximum compression. In this case, the achievable lossless bit rate is governed by a high-order entropy which is less than zeroth-order entropy.

The term *entropy coding* refers to the use of a variable length code to losslessly represent a sequence of symbols from a discrete alphabet. The lower bound or the minimum achievable average rate for memoryless source is the entropy of the source. A practical entropy code must be uniquely decodable, so that there is only one possible sequence of codewords for any unique input sequence. Currently, three popular entropy coding techniques, Huffman coding, Golomb-Rice, and arithmetic coding, are used in modern compression systems and standards [9-27]. We only discuss arithmetic coding in the following section, because it is used in our work.

2.1.3. Arithmetic Coding

In arithmetic coding[28-30], a sequence of symbols is uniquely encoded as a value, which is an interval in the range [0,1). Because the number of values in the unit interval is infinite, it should be possible to assign a unique subinterval to each distinct sequence of symbols. The size of this subinterval is determined by the cumulative distribution (*cdf*) of the random variable associated with the source [1].

Arithmetic coding can be illustrated with an example. Consider a source with a three symbol alphabet, denoted as {a, b, c}, with symbol probabilities as defined in Table 2.1

When we encode a sequence, the subinterval that represents the whole sequence is getting narrowed with respect to each symbol probability. Suppose that the sequence *abacb* is to be encoded. The first symbol, *a*, falls in the interval of [0, 0.6). After *a* is encoded, the low end and high end of the interval become 0 and 0.6 separately. The next interval is defined by subdividing [0, 0.6) in proportion to the probability of the next symbol *b*, according to Table 2.1. Instead of [0.6, 0.7) with respect to the unit interval, the next interval is [0.6x0.6, 0.7x0.6). Applying this procedure will further restrict the range to [0.36, 0.42).

Table 2.1 Probabilities and intervals associated to three symbols of a source

Symbol	Probability	Interval
a	0.6	[0,0.6)
b	0.1	[0.6, 0.7)
c	0.3	[0.7,1)

This process continues for successive symbols, so that the sequence *abacb* is represented by the final interval [0.3852, 0. 39168). The intervals are shown in Figure 2-1, where the size of the interval has been scaled so that the small intervals are visible.

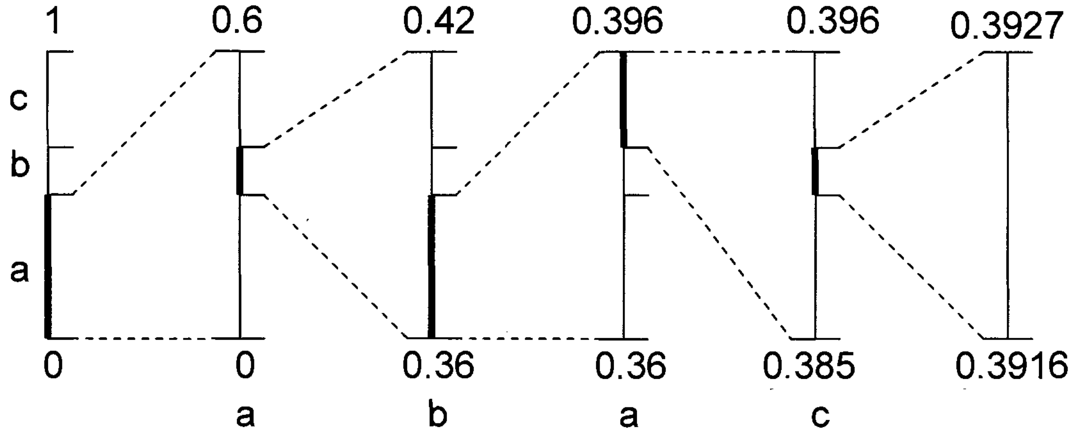


Figure 2-1 Interval for the sequence a b a c b

The maximum number of bits required to encode the interval is:

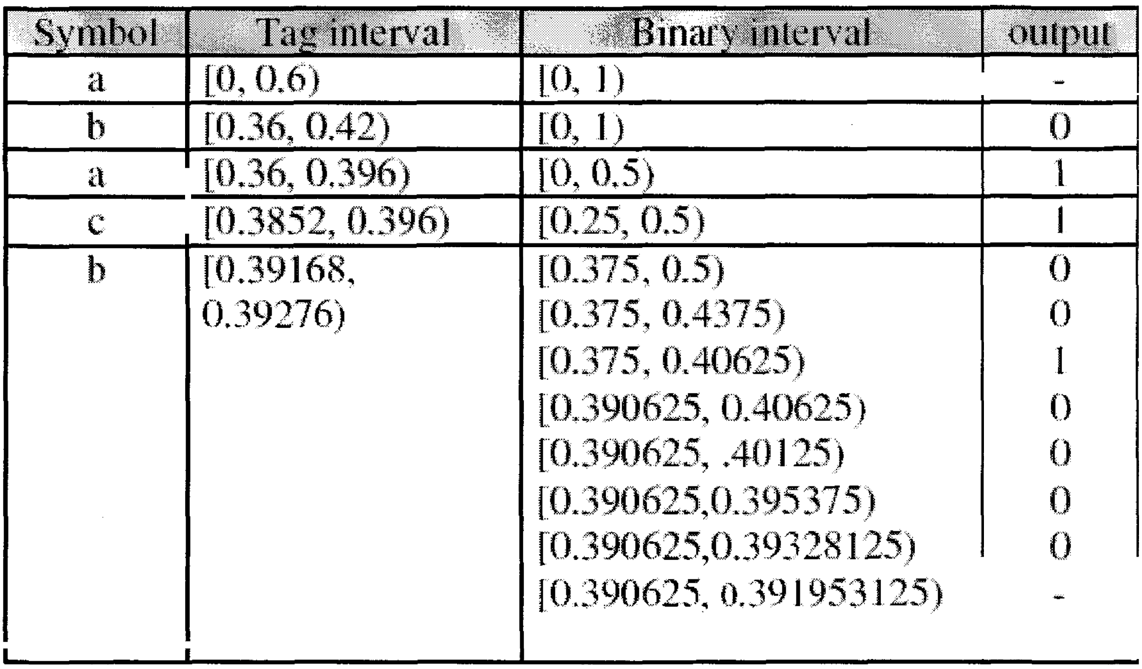
$$ceil(\log(1/p(x_i)))+1 \qquad (2.3)$$

where $p(x_i)$ is the probability of the sequence [1]. In this example, to encode the whole sequence, we need

$$ceil(\log(\frac{1}{p(a)p(b)p(a)p(c)p(b)}))+1 = ceil(8.38)+1 = 10 \ \ bits \qquad (2.4)$$

The interval representing a sequence is coded as a string of bits which identify the tag. The binary bits are sent in the order of precision from the most significant bit. In the example, the first interval [0, 0.6) is not confined to either the upper or lower half of the unit interval, so no bits are transmitted for the symbol *a*. The second symbol *b* restrains the interval between 0.36 and 0.42, which is included in the lower half of the unit interval, so a bit "0" is sent. The third symbol "*a*" constrains the tag to [0.36, 0.396), which falls in the upper half of the interval [0, 0.5), so a bit "1" is sent. The same process continues until the last symbol *b* is encoded. Table 2.2 illustrates the procedure of encoding, showing the transmitted bits, not including the termination bits, and the intervals that they are assigned.

Table 2.2 Encoding the sequence a b a c b



Symbol	Tag interval	Binary interval	output
a	[0, 0.6)	[0, 1)	-
b	[0.36, 0.42)	[0, 1)	0
a	[0.36, 0.396)	[0, 0.5)	1
c	[0.3852, 0.396)	[0.25, 0.5)	1
b	[0.39168,	[0.375, 0.5)	0
	0.39276)	[0.375, 0.4375)	0
		[0.375, 0.40625)	1
		[0.390625, 0.40625)	0
		[0.390625, .40125)	0
		[0.390625,0.395375)	0
		[0.390625,0.39328125)	0
		[0.390625, 0.391953125)	-

In the decoding process, first we meet “0”, which constrains the interval to [0,0.5). We can obtain the first symbol **a**, whose probability range is from 0 to 0.6, completely containing the binary interval [0,0.5). The second binary bit, “1”, narrows the interval to [0.25,0.5), which can not tell us exactly what the second symbol is. The third binary bit “1”, continues to refine the interval to [0.375,0.5), which still does not define the second symbol of the sequence. After the fourth bit “0”, we obtain the second symbol, **b**. We go on decoding with the same logic and finally get **a b a c b** at the end. Table 2.3 describes the decoding procedure.

In the above example, we assume that both the encoder and decoder know the length of the message so that the decoder would not continue the decoding process forever. Otherwise, we need to include a special terminating symbol so that when the decoder sees this symbol, it stops the decoding process.

Table 2.3 Encoding the sequence a b a c b

Input	Binary interval	Decoded Symbol
0	[0, 0.5)	A
1	[0.25, 0.5)	-
1	[0.375, 0.5)	-
0	[0.375, 0.4375)	-
0	[0.375, 0.40625)	B
1	[0.390625, 0.40625)	-
0	[0.390625, 0.40125)	-
0	[0.390625, 0.395375)	A
0	[0.390625, 0.39328125)	C
0	[0.390625, 0.391953125)	B

In summary, the encoding process is simply to narrow the interval of possible numbers with every new symbol. The new range is proportional to the predefined probability attached to that symbol. The output of the encoder is binary bits determined by the sequence tag and the incrementally finer binary intervals with respect to each output. Conversely, decoding is the procedure where the binary interval is narrowed by the input bits, and each symbol is extracted according to its probability and the binary interval.

2.1.4. Adaptive Arithmetic Coding

We have seen how arithmetic coder works when the distribution of the source is available. In many applications the source distribution is not known *a priori*. It is a relatively simple task to modify the algorithm discussed so that the coder learns the distribution as the coding progresses. A straightforward implementation is to start with a count of 1 for each symbol in the alphabet. We need a count of at least 1 for each symbol. If not we will have no way of encoding the symbol when it is first encountered. This assumes that we know nothing about the distribution of the source. If we do know something about the distribution of the source, we can let the initial counts reflect our knowledge.

After the coding is initiated, the count for a symbol is incremented each time it is encountered and encoded. The cumulative count table is updated accordingly. This updating takes place at both the encoder and decoder to maintain the synchronization between the two.

2.1.5. Context Based Adaptive Arithmetic Coding

As mentioned in Section 2.1.2, “unconditional” entropy coding can obtain a lossless coding rate that approaches the zeroth-order entropy of the input source. Given a finite source $X_1, X_2, ..., X_n$, compressing this sequence losslessly requires one to process the symbols in some order and try to estimate the conditional probability distribution for the current symbol based on the previously processed symbols [31]. If we use conditional probabilities, we can do better than the zeroth-order entropy. The minimum code length of the sequence in bits is

$$-\log_2 \prod_{i=0}^{n-1} p(X_{i+1} \mid X_1,\ldots,X_i). \qquad (2.5)$$

The design objective of an adaptive arithmetic coder is to attain a code length approaching the source entropy given above. Given the numerical precision of specific coder implementation (more than sufficient for modern computers), the performance of an arithmetic coder is solely determined by the context model that drives it. The role of context modelling is to estimate the conditional probabilities $p(X_{i+1} \mid X^i)$ where $X^i = X_1,\ldots,X_i$ is the prefix or context of X_{i+1}. Indeed, given a model class, the order of the model or the number of model parameters needs to be carefully selected so as not to negatively impact the code length. If the model order is too low, the true distribution will not be well approximated, while if the model order is too high, the model parameters will not be well estimated. In the literature, this problem is addressed in various ways. The pioneer solution to the problem is Rissanen's algorithm Context [32], which dynamically selects a variable-order subset of the past samples in X^i, called the context C_i. The algorithm structures the contexts of different orders by a tree and it can be shown to be, under certain assumptions, universal in terms of approaching minimum adaptive code length for a class of finite-memory sources. A more recent and increasingly popular coding technique is context tree weighting [33]. The idea is to weight the probability estimates associated with different branches of a context tree to obtain a better estimate of $p(X_{i+1} \mid X^i)$. Because the estimation error decreases with increasing data length, in the limit both the estimation and approximation error can be made to go to zero by increasing the model complexity at the proper rate. This is the basis for universal coding. The Context algorithm and context tree weighting can be shown to be universal for the class of finite-state Markov (FSMX) sources.

Although the tree-based context modelling techniques have had remarkable success in text compression, applying them to image compression poses a great challenge. The context tree can only model a sequence not a two-dimension signal like images. It is possible to schedule the pixels (or transform coefficients) of an image into a sequence so that context tree weighting algorithm can be applied [34-36]. In particular, Mrak *et al.* investigated how to optimize the ordering of the context parameters within the context trees [36]. But any linear ordering of pixels will inevitably destroy the intrinsic two-dimensional sample structures of an image. This is why most image/video image compression algorithms choose *a priori* two-dimensional context model with fixed complexity, based on domain knowledge such as correlation structure of the pixels and typical input image size, and estimate only the model parameters. For instance, the JBIG standard for binary image compression uses the contexts of a fixed size causal template [37]. The actual coding is implemented by sequentially applying arithmetic coding based on the estimated conditional probabilities.

Learning the conditional probabilities $p(X_{i+1} \mid X^i)$, or equivalently $p(X \mid C)$, on the fly using count statistics of the input can be slow in converging to the source statistics. The compression performance degrades when the length of input data is short relative to the size of the context model. In image/video compression the problem is aggravated by the fact that image signals contain long memory (even after a decorrelation transform). A high-order context model is thus required by arithmetic coder to approach the entropy of the image source. Since the number of conditioning states increases exponentially in the order of the context model, the amount of sample data available per conditioning state is diluted exponentially on average, causing the well-known problem of context dilution.

A common practical technique to overcome the difficulty of context dilution is context quantization [2] [38-43]. The idea is to reduce the number of conditioning states by merging those of similar statistics into one. For example, the lossless image compression algorithm CALIC [26] and the JPEG 2000 entropy coding algorithm EBCOT [44] quantize a context C of order d into a relatively small number *M* of conditioning states. Denote the context quantizer by $Q: E^d \rightarrow \{1,2,\cdots,M\}$. The arithmetic coder is then driven by an estimated $p(X \mid Q(C))$ rather than by an estimated $p(X \mid C)$. But these context quantizers and others used in practical image/video compression methods were designed largely in an ad hoc way.

Greene *et al.* were the first to optimize context quantizers under the criterion of minimum conditional entropy for binary sources such as binary images [45]. If X is a binary random variable, then the probability simplex space for $P(X)$ is one dimensional. This reduces context quantizer design to a scalar quantizer design problem, and consequently the problem can be solved by dynamic programming. The same design problem was investigated by Forchhammer *et al.* but for the objective of minimizing the actual code length of adaptive arithmetic code [40]. Large coding gains were made by their design algorithm on MPEG 4 binary mask image sequences.

Recently, some authors including us proposed context quantizer design algorithms that work directly in the context space E^d i.e., the vector space of conditioning events [2] [41-42]. These algorithms are essentially vector quantization (VQ) approach [46] that clusters raw context instances of a training set using Kullback-Leibler distance as the VQ distortion metric. The context quantizer design is done by some variant of the generalized Lloyd method of gradient descent, and consequently the solution is only locally optimal. A daunting and unresolved operational difficulty for this approach is the high description complexity of quantizer mapping function (inverse quantization function). The quantizer cells in the context space are generally not convex, and even consist of disjoint regions [39]. This makes the decoder implementation unwieldy, requiring a huge look-up table.

To circumvent the problem of inverse quantization in context space, Wu proposed a different context quantizer design technique[38] [47-49], which actually predated all other VQ-based context quantization methods. It first performs a principal component transform of the context space and then forms a convex partition of the context space in the principal direction under the criterion of minimum Kullback-Leibler distance. This practical technique was successfully applied to Golomb-Rice coding of 3D wavelet coefficients for volumetric medical image compression [50] and to adaptive arithmetic coding for high-performance lossless image and video compression [51]

Up to now all the context quantizers are optimized with respect to the statistics of a training set. An ensuing question is how to handle any mismatch in statistics between the training set and an input image. This problem has remained largely untreated.

2.2. Quantization

2.2.1. Quantization Basics

Quantization [46] is the act of mapping a large set of different values to a smaller set, which is one of the basic ideas of lossy data compression. Figure 2-2 is an example of a scalar quantizer, where all the real values in the *x* axis are mapped into only six values in the *y* axis. In this example the values that reside in the range $[0, \Delta)$ are mapped into $\Delta/2$, etc.

Usually, each of the values on the *y* axis is assigned a quantization index, and the indexes are entropy coded at the coder side. At the decoder, first the indexes are entropy decoded, and then since the exact values of the coded samples are not known, each index is mapped into a reconstruction value that in some sense, optimally represents the samples in that interval. This mapping of index to value is usually predetermined and the encoder uses this to decide on the quantization index. Also since the exact value of the sample cannot be recovered at the decoder, the resulting compression will be lossy.

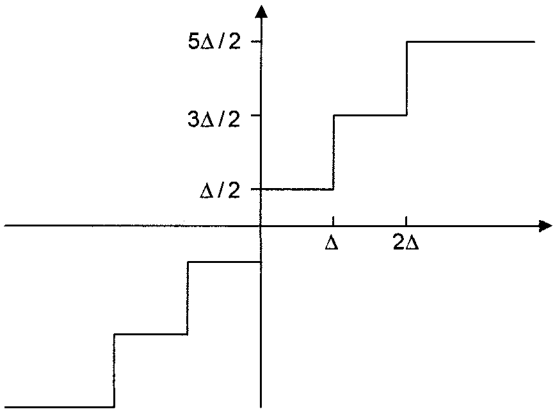


Figure 2-2 Example of a uniform scalar quantizer

The design parameters in every scalar quantizer include the sizes of each interval in both of the *x* and *y* axes, the number of the quantization levels and the reconstruction values. The design, in turn, depends upon the statistics of the source samples, and the conditions and constraints that exist in each practical problem.

2.2.2. Adaptive Quantization and Classification

Chrysafis and Ortega [52] presented a novel approach that combined context classification and adaptive quantization together in the coding of the image subbands. They used the wavelet and applied a uniform threshold quantizer on the subband coefficients. For each symbol, a prediction is made using the nearest three “causal” symbols and one parent-band symbol. An Entropy Constrained Scalar Quantizer (ECSQ)1 is used on the predictor to classify the current pixel. The number of output points of the quantizer, the context size, is 11 in their experiment [52]. This backward adaptive classification technique, which decides each pixel's context state determines the probability model for the arithmetic coder. The adaptive quantization in ECSQ is implemented with respect to the rate distortion criterion, $J = D + mR$, where J is the rate distortion cost, D and R represent the distortion and the rate needed respectively, and m is a Lagrange multiplier, depending only on the statistics of the image. In this algorithm, the context information includes the quantized coefficients not only in the same subband, but also in its parent subband as well.This context model tries to give a more precise predictor for the subband coefficient.

Yoo, Ortega and Yu [53] gave a different consideration for the quantizer sets in their work. They couple the context classification and the quantization techniques in their algorithm. First, they separate the subband coefficients into different classes; then they apply different quantization to each class using a bit allocation strategy. The activity of the current coefficient is predicted by a weighted average magnitude of six previously transmitted quantized coefficients. The current pixel is classified by thresholds on the estimated predictor. Unlike the work by Chrysafis [52], the classification thresholds are designed in an iterative procedure aiming at maximizing the coding gain due to classification. The iterative merging algorithm, which tries to merge the pair of classes with the smallest gain, converges to a local optimum at each iteration that increases the classification gain. A special class called "zero context" was adopted to separate this kind of context (consisting of all zero-quantized coefficients which contain little information for estimation) from the others. The classification can be formed on a coefficient-by-coefficient basis that overcomes the shortcomings of block-based classification. This classification can split subband coefficients into classes of different variances and has the advantage of achieving classified regions with arbitrary shapes. After the classification, the quantizer is applied to each classified subband coefficient. Under the assumption of a Laplacian distribution model, an adaptive Uniform Threshold Quantizer (UTQ) can be derived from the online estimation of model parameters within each class.

CHAPTER 3.
CONTEXT QUANTIZATION FOR ENTROPY CODING OF NON BINARY SOURCE

3.1. Context Quantization

3.1.1. Problem Formulation

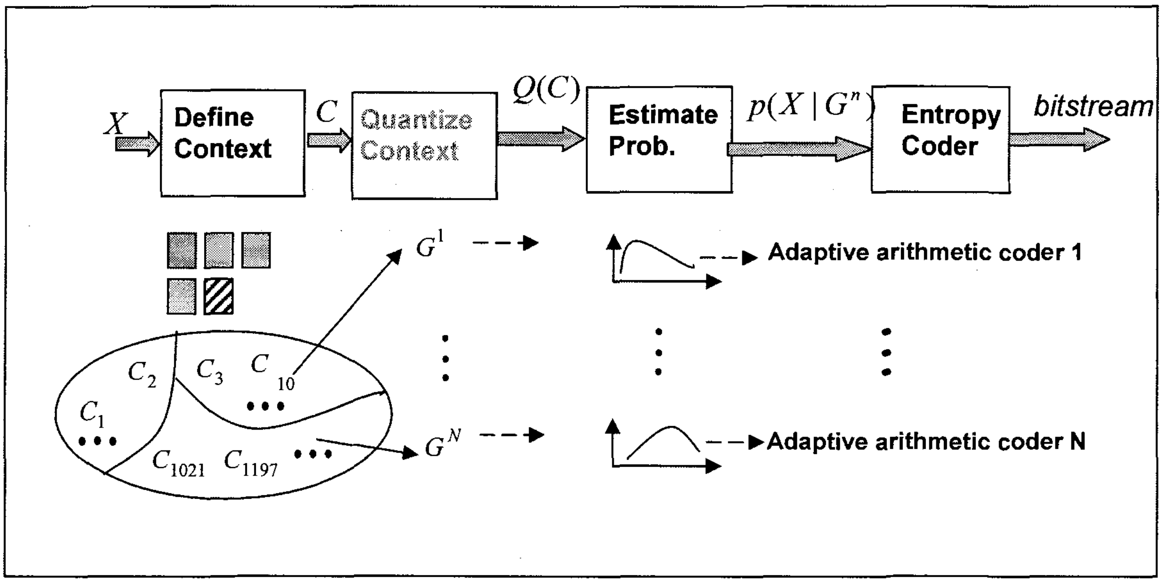


Figure 3-1 Context Quantization

As discussed in 2.1.5, an adaptive arithmetic coder first classifies the current data into a coding state, and then compresses the data using an estimated conditional probability for the coding state. Correspondingly, the more precisely the coding states distinguish different source distributions, the more efficient the coder will be. Therefore, the key to high coding performance is how we classify the data, in other words, how to define the coding context.

Note that only causal context model is studied in our work. A causal context model uses a combination of "past" values to form the context. In a raster scanned image, the causal context model contains the neighbors to the left and at the top of the symbol being coded. As shown in Figure 3-1 the context template contains the north, northwest, northeast and west pixel of the pixel being coded. No side information is required to decode the sequence of bits, because the decoder has reconstructed the previous symbols to obtain the context of the current decoded symbol.

Consider a source X with K different symbols. Given a defined casual context template, the context space is composed of all the possible context instances. For example, if the context template is defined as four casual neighbours as shown in Figure 3-1, the corresponding context space contains K^4 possible context instances (pixel patterns). However, many of the context instances may not appear in a particular source, the actual context space size *M* may then be much smaller than the maximum size. The source data is then classified into *M* states. In traditional context-based adaptive arithmetic coder, the conditional probability for each state is estimated on the fly and used to generate the code stream. When *M* is large, many context instances rarely occur in a particular image, sometimes only happen once. The amount of data for learning will be small for these context instances, which causes severe estimation error and consequently leads to poor compression performance. This is the well-known problem of context dilution.

The solution to the context dilution problem is context quantization. The context space will be partitioned into N subsets. The context instances in each subset will be merged into a coding state in which the data share the same probability model when being compressed by an arithmetic coder. Because the amount of data for learning the statistics of the merged state increases, more accurate probability model will be used to drive the arithmetic coder. However, the decrease of the number of coding states may cause an increase in the code length . Therefore, the partition needs to be optimized. Intuitively, context instances with similar probabilities should be grouped together. Then the questions become how to measure the similarity between the probabilities and how to find the optimal partition in the sense of achieving the balance between the benefit of using many conditioning states to lower the entropy and the cost associated with context dilution.

3.1.2. Histogram Quantization

Our approach is based upon the notion of a histogram quantizer. It takes any input histogram which corresponding to a context instance and matches it to a finite set of histograms from a "codebook". More precisely, let $T = (T_1, T_2, ... T_K)$ be one of *M* conditional probability histograms for a source X having *K* different symbols, with T_k being the conditional probability of symbol *k*. An *N*-state histogram quantizer is a mapping that assigns to each input histogram, *T*, a reconstruction histogram, $T^{'} = q(T)$ that is drawn from a finite-size codebook of *N* histograms, $A_N = \{R^i, i = 1,..., N\}$, where R^i denotes the i-th target histogram (a codeword in the histogram codebook). The quantizer, *q*, is completely described by two elements: the reconstruction alphabet *A* and the partition of the input histogram space. This partition is defined by the set $S = \{S_i, i = 1,\ldots,N\}$, with $S_i = \{T : q(T) = R^i\}$.

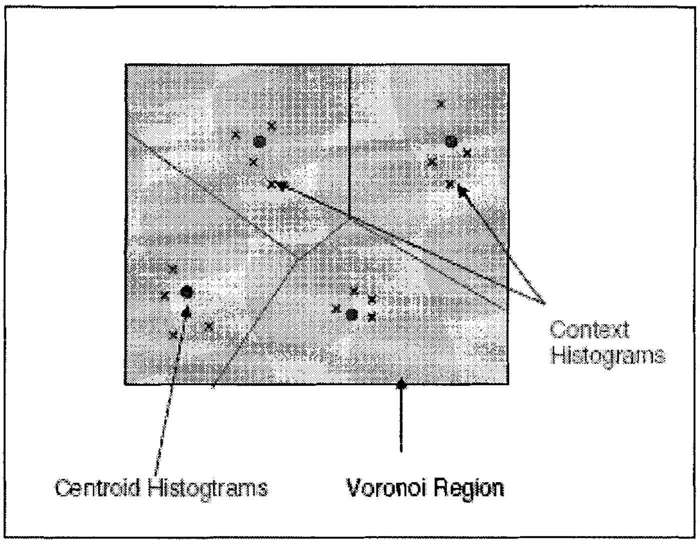


Figure 3-2 Illustration of Histogram Quantization

In data compression practice, the design goal of context quantization (histogram grouping) is to achieve the minimum arithmetic code length. This is equivalent to, as we will discuss in subsections 3.1.3 and 3.1.5, minimizing the relative entropy or Kullback-Leibler distance between the input histogram *T* and its quantized version R^i, which is defined by

$$d(T,R^i) = H(T \| R^i) = \sum_{k=1}^{K} T_k \log(T_k / R_k^i) \qquad (3.1)$$

Although relative entropy can be viewed as a distance measure between two distributions, it is not a true distance metric. It is not symmetric nor satisfies the triangular inequality. Nevertheless, $d(T,R^i)$ specifies the increase in bit rate if one uses the histogram R^i to code a source of the histogram *T*.

Context quantization is a problem of vector quantization (VQ). However, unlike VQ of signals, context quantizer works in a probability (histogram) space instead of a sample vector space. The VQ interpretation of context quantization can be seen in Figure 3-2, in which the crosses represent conditional probability histograms and the red dots are the centroid histograms. In optimal design of context quantizers, we apply the classic VQ design approach of gradient descent (commonly known as the generalized Lloyd algorithm) in the probability space and use relative entropy as the "distortion" measure The design algorithm is formally stated as the following.

1) Given the desired number of quantized states in the context space, *N*, start with an initial reconstruction histogram codebook, $A_N^{(0)}$; average distortion $D^{(0)}$ and iteration $m=0$. Select ε.

2) At interation m, determine the *N* quantizer cells defined by

$$S_i^{(m)} = \{T : d(T,R^i) < d(T,R^j), \forall j \neq i\}, \qquad i = 1,\dots,N \qquad \textbf{(3.2)}$$

and compute the average distortion, $D^{(m)}$, between the input and target histograms as

$$D^{(m)} = \sum_{i=1}^{N} D_i^{(m)} \qquad (3.3)$$

$$D_i^{(m)} = \sum_{T \in S_i^{(m)}} \frac{p(T)}{p(S_i^{(m)})} d(T, R^i) \qquad (3.4)$$

Where $p(T)$ is the probability of histogram and $p(S_i^{(m)})$ is the sum probability of all the histograms in the cell S_i .

3) Stop if $\left|D^{(m-1)} - D^{(m)}\right| / D^{(m)} < \varepsilon$

4) Determine the codebook for iteration $m+1$, $A_N^{(m+1)}$, by computing the average histogram for each $S_i^{(m)}$; this is done element by element according to

$$R_k^i = \sum_{T \in S_i^{(m)}} \frac{p(T)}{p(S_i^{(m)})} T_k \qquad i = 1, \ldots, N \qquad (3.5)$$

5) $m = m+1$, go to step 2).

3.1.3. Convergence of Context Quantizer Design Algorithm

In order to guarantee the convergence of the algorithm, we require that $D^{(m-1)} - D^{(m)}$ be non-negative. It is clear that step 2) above is a nearest neighbor calculation and that it can only lower the distortion; however, we need to prove that step 4) also reduces the distortion. Since the total distortion is made up of a sum of the D_i terms, we can treat them individually. Expanding (3.4) using (3.1) gives

$$D_i^{(m)} = \sum_{T \in S_i^{(m)}} \frac{p(T)}{p(S_i^{(m)})} \sum_k T_k \log(T_k / R_k^i)$$

$$= \sum_{T\in S_i^{(m)}} \frac{p(T)}{p(S_i^{(m)})} \sum_k T_k \log T_k - \sum_{T\in S_i^{(m)}} \frac{p(T)}{p(S_i^{(m)})} \sum_k T_k \log R_k^i \qquad (3.6)$$

Changing R_k^i has no effect on the 1st term and we thus minimize $D_i^{(m)}$ by maximizing the 2nd term. Defining

$$W = \{W_k, k = 1,...,N\}; \qquad W_k = \sum_{T\in S_i^{(m)}} \frac{p(T)}{p(S_i^{(m)})} T_k \qquad (3.7)$$

allows us to write the second term as

$$\lambda = \sum_{T\in S_i} \sum_k \frac{p(T)}{p(S_i)} T_k \log R_k^i = \sum_k W_k \log R_k^i \qquad (3.8)$$

Now, inspection shows that both $\sum_k W_k = 1$ and $\sum_k R_k^i = 1$, which means that both W and R^i are valid pmf's. Since the relative entropy between two pmf's is non-negative, we have

$$H(W \,\|\, R^i) \geq 0 \qquad (3.9)$$

Therefore,

$$\sum_i W_k \log(W_k / R_k^i) \geq 0 \;\Rightarrow\; \sum_i W_k \log R_k^i \leq \sum_i W_k \log W_k \qquad (3.10)$$

with equality when $W_k = R_k^i$. We thus see that λ will be maximized if and only if this equality is true. Since this is exactly what step 4) forces, this step can never result in an increased distortion.

3.1.4. The Number of Context Instances

The monotonicity of the objective function as the generalized Lloyd method iterates ensures that we obtain locally optimal *N* coding states. However, there is another design parameter to be determined. That is the optimal number of coding states *N*. The value of *N* governs the trade-off between the accuracy of the quantized histogram and the severity of context dilution. The larger the value of *N*, the finer the classification of different histograms, but more samples are needed to learn the conditional probability. We need to find the optimal value of *N* that achieves minimum code length in conjunction with an optimal partition of probability space.

We approach the above problem using the technique of quantizer cell splitting [46]. As presented below, the number of context instances is constrained to be a power of 2; however, this restriction is easily lifted with trivial modifications.

1) Initialization: Let R^1 be the centroid histogram of the *M* histograms that form the context space. Set $n=1$ and define $A_1 = \{R^1\}$.

2) $n = 2n$. To obtain double the number of contexts, each set S_i split by forming two new "centroids": R^i itself and the histogram in S_i that is closest to R^i.

3) Run the histogram-quantizer algorithm to produce a system with *n* contexts

4) Repeat 2) and 3) until the actual rate goes up due to the context dilution.

3.1.5. Optimality of Proposed Context Quantizer

As stated at the beginning of this chapter, in optimal context quantization our goal is to make the conditional entropy with quantized context $H(X \mid Q(\mathrm{c}))$ as close to the conditional entropy with the original defined context $H(X \mid \mathrm{c})$ as possible. In other words, the optimal context quantizer should minimize the difference between these two conditional entropies $H(X \mid Q(\mathrm{c})) - H(X \mid \mathrm{c})$. A natural question to ask is whether the context quantizer designed using the proposed iterative algorithm will achieve this objective. The answer to this question is yes, as we will show below.

A context quantizer Q partitions a context space into N subsets:

$$G_n = \{\mathrm{c} \mid Q(\mathrm{c}) = n\}, n = 1, \dots, N \qquad (3.11)$$

The associated sets of probability mass functions are

$$B_n = \{p(X \mid \mathrm{c}) \mid \mathrm{c} \in G_n\} \qquad (3.12)$$

The centroid probability mass function of the quantization region B_n is

$$p(x \mid \mathrm{c} \in G_n) = \sum_{\mathrm{c} \in G_n} \frac{p(\mathrm{c})}{p(\mathrm{c} \in G_n)} p(x \mid \mathrm{c})$$

$$= \frac{1}{p(\mathrm{c} \in G_n)} \sum_{\mathrm{c} \in G_n} p(\mathrm{c}) p(x \mid \mathrm{c}) \qquad (3.13)$$

Then,

$$p(\mathrm{c} \in G_n) p(x \mid \mathrm{c} \in G_n) = \sum_{\mathrm{c} \in G_n} p(\mathrm{c}) p(x \mid \mathrm{c}) \tag{3.14}$$

The conditional entropy with the quantized context is

$$H(X \mid (Q(\mathrm{c}))) = -\sum_n \sum_x p\{\mathrm{c} \in G_n\} p(x \mid \mathrm{c} \in G_n) \log p(x \mid \mathrm{c} \in G_n)$$

$$= -\sum_n \sum_x \log p(x \mid \mathrm{c} \in G_n) (\sum_{\mathrm{c} \in G_n} p(\mathrm{c}) p(x \mid \mathrm{c})) \tag{3.15}$$

Since all the contexts in the subset G_n share the same conditional probability, $p(x \mid \mathrm{c} \in G_n)$ can be written as $p(x \mid Q(\mathrm{c}))$. Then

$$H(X \mid Q(\mathrm{c})) = -\sum_n \sum_x \sum_{\mathrm{c} \in G_n} p(\mathrm{c}) p(x \mid \mathrm{c}) \log p(x \mid Q(\mathrm{c}))$$

$$= -\sum_x \sum_n \sum_{\mathrm{c} \in G_n} p(\mathrm{c}) p(x \mid \mathrm{c}) \log p(x \mid Q(\mathrm{c})) \tag{3.16}$$

Applying (3.14), we have

$$H(X \mid Q(\mathrm{c})) = -\sum_x \sum_{\mathrm{c}} p(\mathrm{c}) p(x \mid \mathrm{c}) \log p(x \mid Q(\mathrm{c})) \tag{3.17}$$

The conditional entropy without context quantization is

$$H(X \mid \mathrm{c}) = -\sum_m \sum_x p(\mathrm{c}) p(x \mid \mathrm{c}) \log p(x \mid \mathrm{c}) \tag{3.18}$$

Our goal is to minimize the difference between the conditional entropies before and after context quantization

$$H(X \mid Q(\mathrm{c})) - H(X \mid \mathrm{c}) = -\sum_{\mathrm{c}}\sum_{x} p(\mathrm{c})p(x \mid \mathrm{c})\log p(x \mid Q(\mathrm{c})) + \sum_{\mathrm{c}}\sum_{x} p(\mathrm{c})p(x \mid \mathrm{c})\log p(x \mid \mathrm{c})$$
$$= \sum_{\mathrm{c}} p(\mathrm{c})\sum_{x} p(x \mid \mathrm{c})\log\frac{p(x \mid \mathrm{c})}{p(x \mid Q(\mathrm{c}))} = \sum_{\mathrm{c}} p(\mathrm{c})H(p(x \mid \mathrm{c}) \,\|\, p(x \mid Q(\mathrm{c}))) \qquad (3.19)$$

Apparently, minimization of $H(X \mid Q(\mathrm{c})) - H(X \mid \mathrm{c})$ equals to minimization of the average relative entropy between the conditional probability mass function of each context in the context space and its corresponding quantization value. In other words, we can find a locally optimal context quantizer in the sense of minimizing the conditional entropy with our proposed iterative algorithm.

3.1.6. Experimental Results

We test the proposed context quantization algorithm using two types of sources with memory.

Gauss-Markov Sequence with Flipping Sign

We first test our algorithm on a 1^{st}-order Gauss-Markov source modified to have zero correlation by randomly flipping the sign of each sample with a probability 0.5 after the sequence has been generated. We call this a GM-F source and select it as a test case since we know the correct answer and it demonstrates the power of our approach. Memory without correlation is also common in wavelet-transformed images.

We set the correlation coefficient ρ of GM-F source as 0.9 and generate a 10^7 sample sequence. We then apply a 32-level uniform quantizer whose loading factor f_l is set to 4, a value chosen to balance the overload and granular distortion of the quantizer. The context template is defined as the two previous samples, X_{-1} and X_{-2}, as shown in Figure 3-3. The resulting context space in the generated sequence contains *M*=774 nonzero histograms out of a possible 1,024.

We apply the proposed iterative algorithm on these histograms, with the quantization level starting from $N = 1$. It then increases in power of two. Since the source is 1st-order Markov, all of the dependency should be in X_{-1} . Therefore, we expect no further drop in entropy when $N = 32$ because there are 32 possible values for X_{-1}. However, the sign flipping operation removes any sign distinction. As a result, 16 distinct contexts should be sufficient to describe the source. The shape of all these 16 conditional probability histograms will be bimodal with the two peaks sitting symmetrically on the positive and negative side, as shown in Figure 3-4. That means the conditional probability histograms are only based on the magnitude of X_{-1}.

The experimental results for the GM-F source with $\rho = 0.9$ are shown in Figure 3-5. It is indeed seen that there is little point in using more than 16 contexts. We can get more information of what is happening by looking at the conditional histograms themselves and these are shown in Figure 3.5 in the case of $N = 16$. As expected, the histograms are all bi-modal. Indeed, the curves in the figure are essentially identical to the conditional histograms based only on the magnitude of the previous sample.

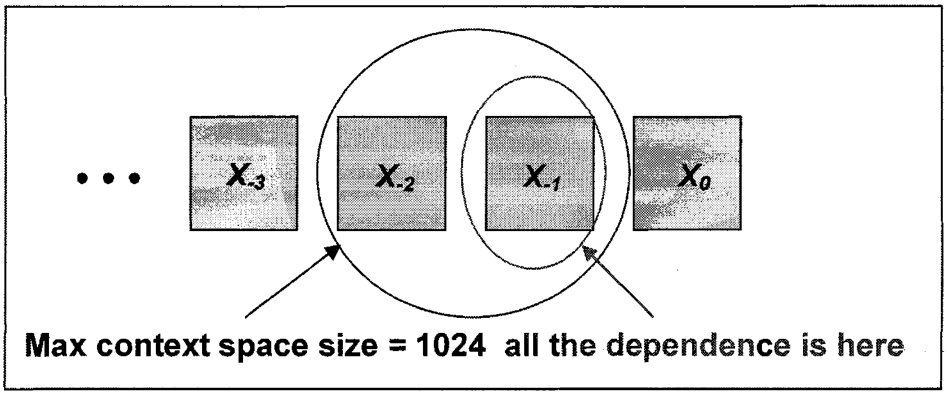


Figure 3-3 Context Definition of GM-F Source

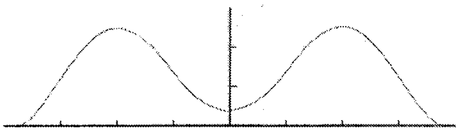


Figure 3-4 Expected Individual Converged Conditional Probability Histogram

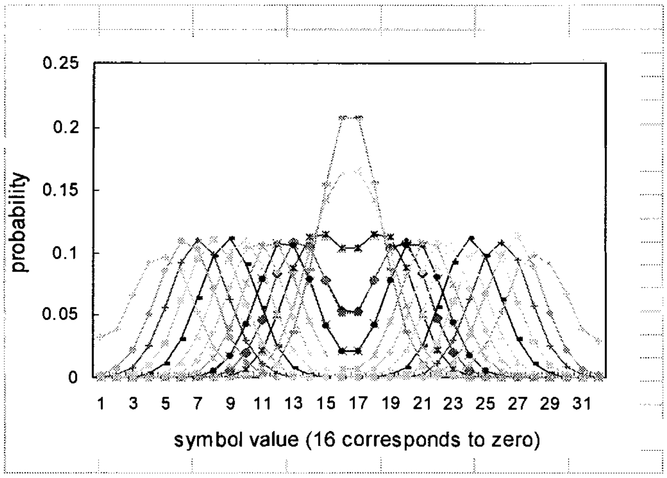
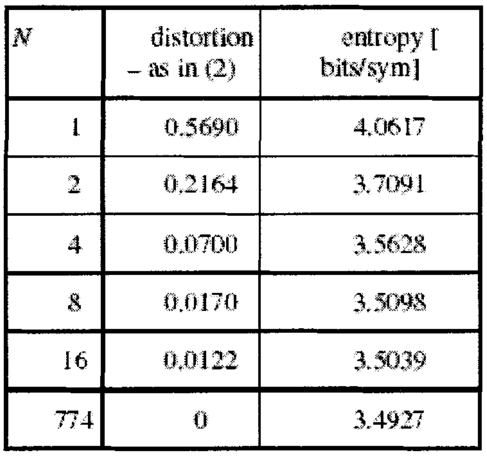


Figure 3-5 Actual Converged Conditional Histograms for GM-F Source

Table 3.1 Context Merging for the GM-F Source



N	distortion – as in (2)	entropy [bits/sym]
1	0.5690	4.0617
2	0.2164	3.7091
4	0.0700	3.5628
8	0.0170	3.5098
16	0.0122	3.5039
774	0	3.4927

Wavelet Subband Images

The second source is an image processed by a three level wavelet transform as shown in Figure 3-6. Five 512x512 gray scale images are used to test the performance of our scheme. The filter set is the standard 9-7 configuration [54]. In this case, we have no idea of the number of the optimal context quantization level and the shape of the converged conditional probability histograms. As with the previous source, we quantize the data with a uniform quantizer and vary N in an identical pattern. Quantizers are designed for each subband by determining the difference between the maximum and minimum values of the coefficients and dividing this number by 16, the desired number of quantization levels. This last number is set fairly arbitrarily since our focus is on the entropy coding. The context template is defined as the four causal nearest neighbors, resulting in a raw count of 313,776. However, most of them have empty histograms.

Figure 3-7 to Figure 3-11 plot entropy as a function of N for 10 different subbands of five wavelet images. We can see that the estimated conditional entropy decreases with increasing N; however, we also see that the difference between the estimated conditional entropy and the true rate obtained from the arithmetic coder with the proposed context quantization method is increasing due to context dilution. Considering LL_3 of the image Barbara specifically, we see that context dilution begins to have a serious effect when $N = 128$. At this point, the overall rate begins to rise quickly from its lowest value of 1.02 bits/symbol. We found that there are 3020 contexts with non-zero histograms in the context space. Using this number of contexts with a real arithmetic coder resulted in a rate of 4.51 bits/symbol, compared with an "ideal" conditional entropy of 0.46 bits/symbol.

3.1.7. Conclusions

In this section we presented a context quantization method for adaptive arithmetic coders. Our method employs a histogram quantizer to partition the context space into the desired number of subsets. Similar to VQ, a splitting algorithm is used for initialization. Our experiments have showed that our method has the potential to automatically discern hidden structure in data and is able to find a (locally) optimal context quantizer in the sense of minimizing the conditional entropy. Our method can be applied to other entropy coding schemes and improve the overall compression efficiency.

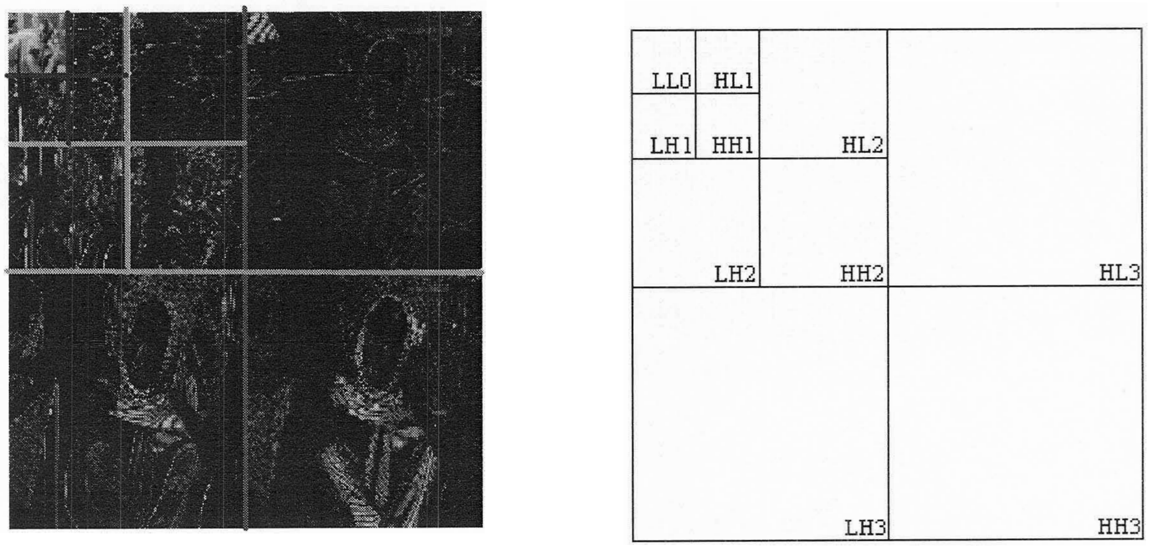


Figure 3-6 The 3-scale Wavelet Transform

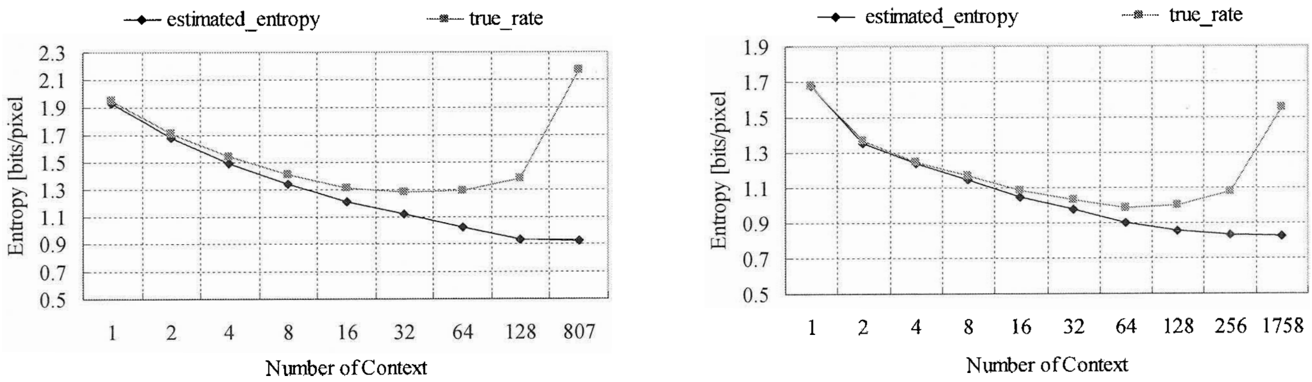


Figure 3-7 Barb Subbands HL_1 **and** HL_2

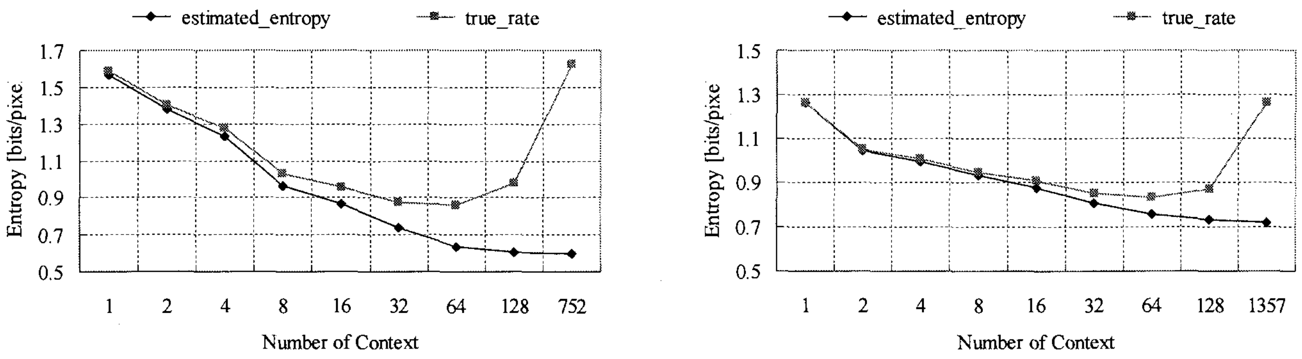


Figure 3-8 Goldhill Subbands HH_1 **and** HL_3

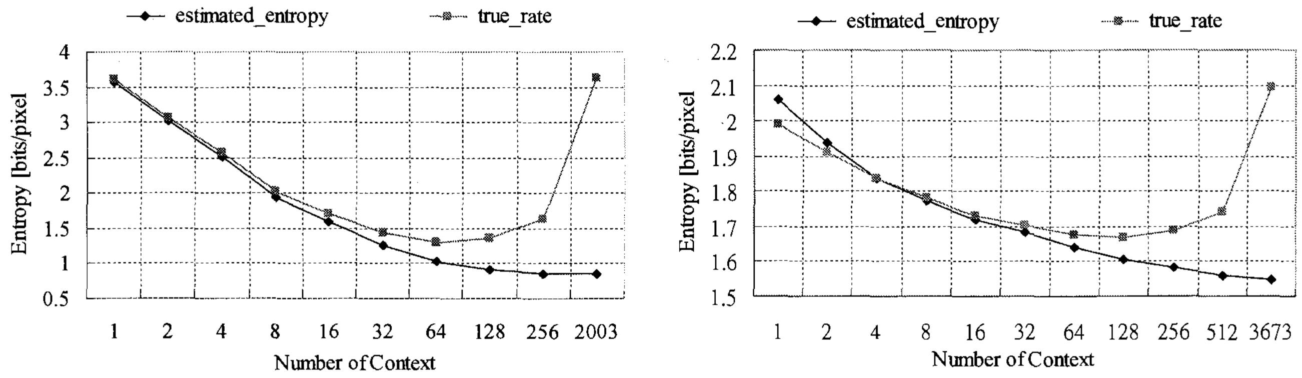


Figure 3-9 Baboon Subbands LL_0 **and** LH_3

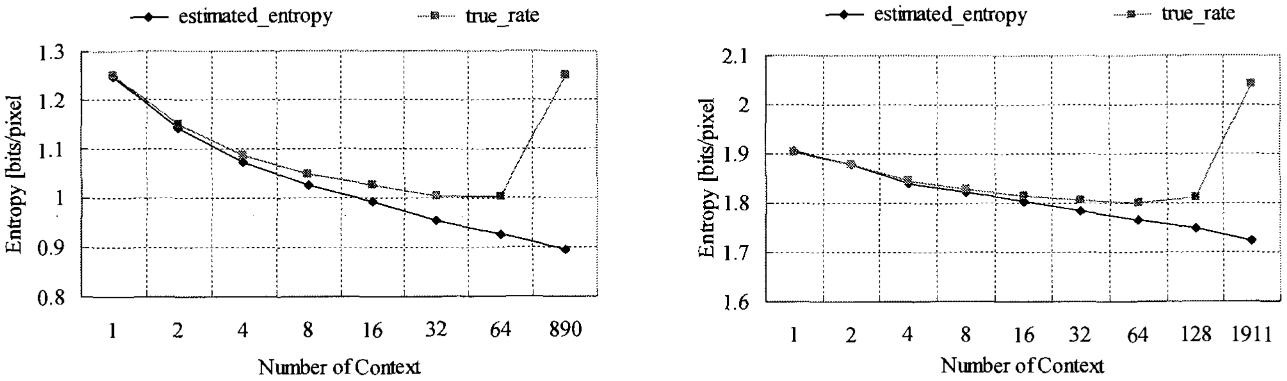


Figure 3-10 Peppers Subbands LH_2 and HH_3

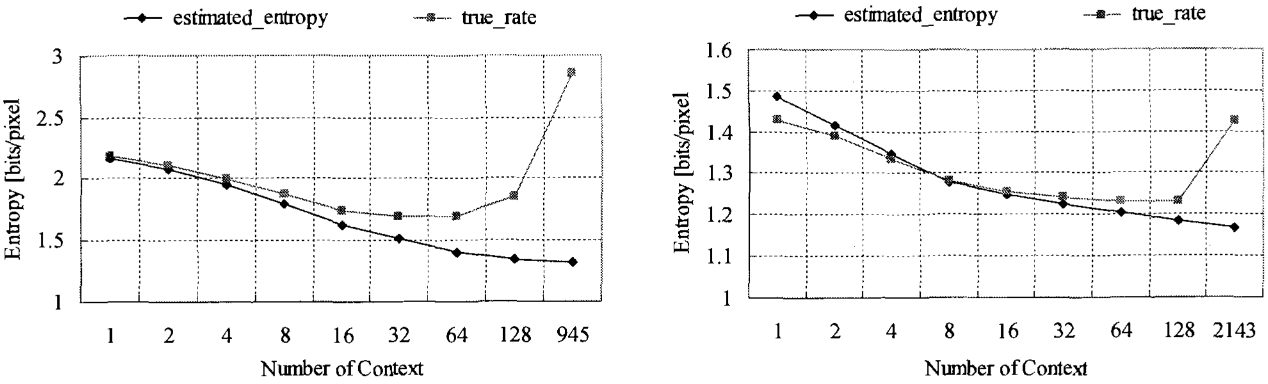


Figure 3-11 Lena Subbands LH_1 and HH_2

3.2. Context Quantizer Description

3.2.1. Motivation

In the previous section, we proposed a context quantization method that uses a clustering procedure to reduce a large set of context instances to a manageable number of coding states. This method is proved to be locally optimal in the sense of minimizing the conditional entropy given a fixed context template. However, if the class definitions are made image adaptive, the context quantizer mappings must be known to the decoder and transmitted as side-information. Therefore the resulting actual code length will be the sum of the bits of encoding the input data using the proposed context quantization scheme and the side information for the description of the context quantizer mappings.

Of course, an alternative to using side information is to use training data to design an off-line optimized fixed context book. However, training set is a solution that can not be well fitted to single image characteristics, which will result in a large reduction of the coding efficiency in the case of mismatch statistics. This problem will be more severe when the number of symbols in the context template is large or/and if the symbol alphabet is large. Furthermore, huge training sets will be needed for those cases.

In this section, we examine two methods for efficiently describing the context book, which completely specifies the partition of the context space. One strategy is to decrease the number of entries in the context book by automatically constructing a new small context space using a metaphor from quantization. The other approach is designed to reduce the bits spent on each individual entry in the context book.

3.2.2. Efficient Context Quantizer Description

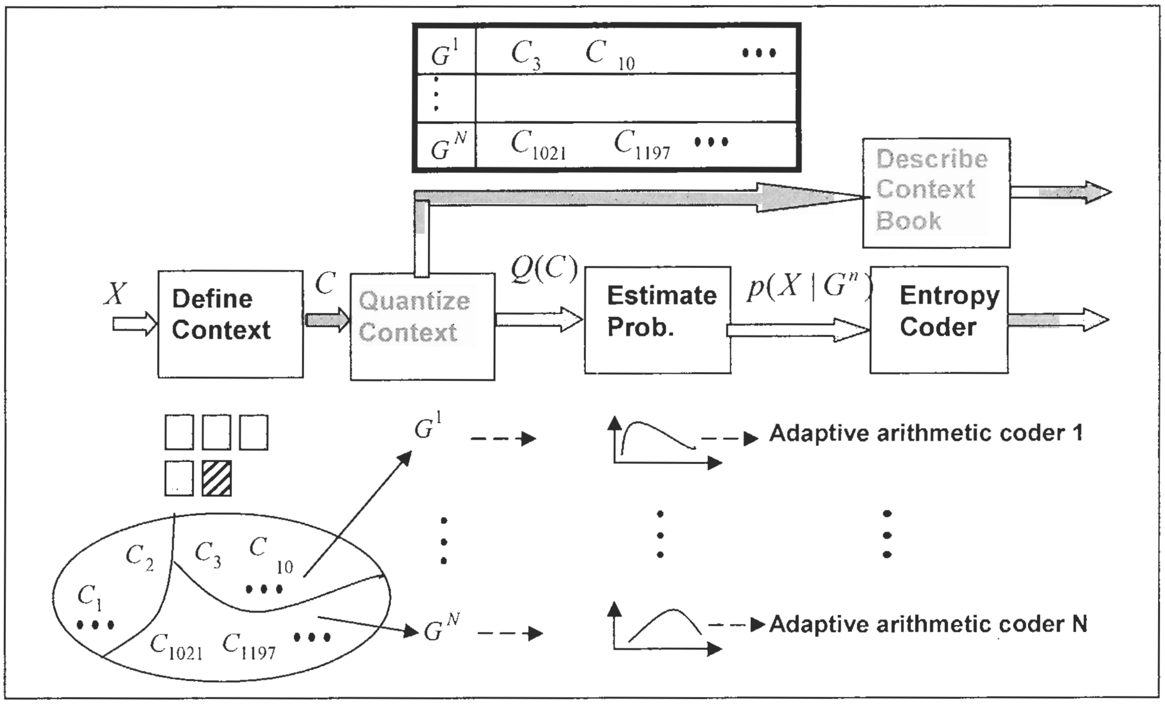


Figure 3-12 Context Quantization and Description

The simplest way of implementing the optimal context quantizer is to use a look-up table. We call this look-up table the *context book*, and organize it into N rows corresponding to the N -partition of the context space. There are a total of *M* different contexts in the context space and each of them belongs to one of N subsets according to the optimal context quantizer designed using the scheme proposed in section 3.1. The context book describes the mappings of these *M* contexts; each row in the context book is a list of all the contexts in one subset. Note that the length of each row is generally not equal, since each row defines a different context subset. Therefore the context book will be a table with *M* entries, which are the context indices organized into N rows. It will be easy to obtain the context quantizer output index if we know the specific context index.

There are two factors that affect the amount of bits required to describe the context book: the number of the entries in the context book, *M*, and the size of an individual entry. The number *M* is quite large for high-order context template, especially in the case of non-binary alphabets. Furthermore, the bits required to specify an individual context index can be significant since the total number of all possible raw contexts is also large. Therefore, it is obviously inefficient, if not impossible, to send the context book directly as side information.

Based on the above observations, two methods, coarse context quantization (CCQ) method and entropy coded state sequence (ECSS) method are proposed to solve the side information problem. CCQ method is aimed at reducing the number of entries in the context book. ECSS method is designed for reducing the bits spent on individual entry.

3.2.2.1. Coarse Context Quantization (CCQ)

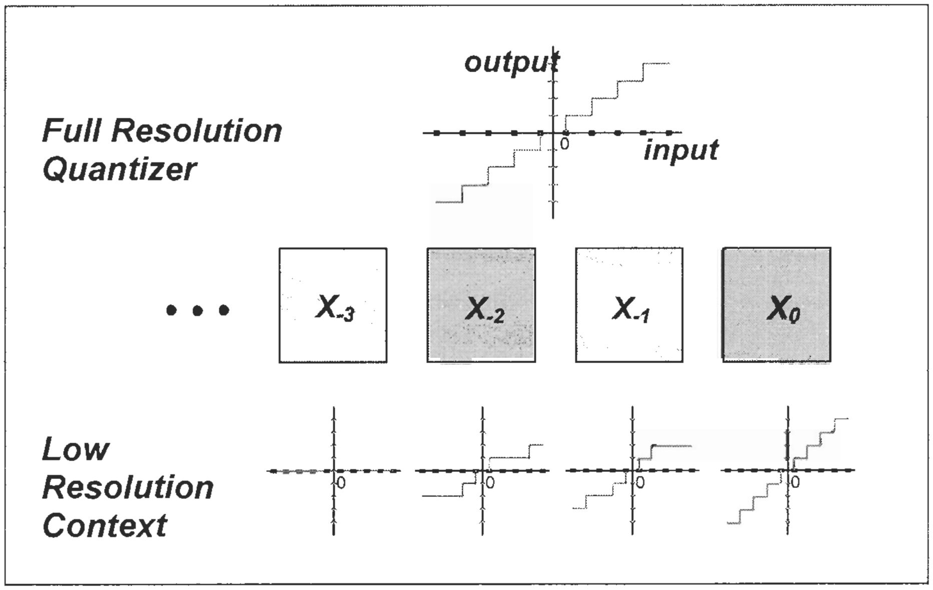


Figure 3-13 Illustration of Coarse Context Quantization

One approach to reducing the side information is to decrease the size of the context space. In this section, we present an algorithm that efficiently lowers the context space size using a metaphor from quantization. In essence, we use a set of coarse quantizers that if applied to the symbols that form the context will automatically construct a new context space with fewer contexts. By doing this, we make certain groupings in the original context space. Some of these groupings will also occur when using the context quantization method described in section 3.1. In this case, we actually describe a fraction of the context book using a set of coarse quantizers, which is obviously efficient. However, applying CCQ also produces some other groupings that do not occur when applying our proposed context quantization method on the original context space, which means that these groupings are not part of the locally optimal groupings in the sense of minimizing the conditional entropy. Nevertheless, we may still allow some of these non-optimal groupings to occur if they do not result in a large bit-rate penalty for the data-stream. In other words, we are making a trade-off between the data and the side information in order to minimize the total bits spent to represent both.

We now seek to reduce the size of the context space through the use of structure. Specifically, we requantize the symbols in the specific context template in order to reduce their precision. Our task is to find the set of quantizers (different ones for each symbol) that will not dramatically increase the data rate. In effect, we generally want finer quantizers to be used for the context samples that are closely related to the sample being coded. Coarse quantizers should be used for those symbols which are almost independent of the target symbol. The basic idea of coarse context quantize is illustrated in Figure 3-13.

We start with a set of full resolution quantizers for all the context pixels defined in the context template. We then pick one context pixel and sequentially erase the boundaries between the quantization levels. After each erasure we evaluate the cost in units of bits associated with the erasure. If the cost is more than a threshold, ε, then we define the logical value of this boundary, b_m, to *true*, if the cost is less than ε, then we erase the boundary and set $b_m = false$.

The cost is determined by the difference between F_m, the conditional entropy when the boundary is erased and H_m, the conditional entropy of the previous stage, when the last boundary is checked. The new rate H_m, will be F_m if the cost is tolerable; otherwise it will be H_{m-1}. If each source symbol takes on L possible values and the size of the context template is K, then for each of the context symbols we will have $L-1$ boundaries to check. The total number of boundaries is thus $P = K(L-1)$.

The algorithm is formally stated as the following:

1) Initialization: Calculate H_0 , the conditional entropy under the definition of the original context space; m = 0; set the boundary information $b_i = true$, for $1 \leq i \leq P$; select ε.

2) Set $b_m = false$. Construct the new context space and calculate F_m , the conditional entropy when the corresponding boundary is erased.

3) If $F_m - H_{m-1} \leq \varepsilon$ then

   $b_m = false$

   $H_m = F_m$

   else

   $b_m = true$

   $H_m = H_{m-1}$

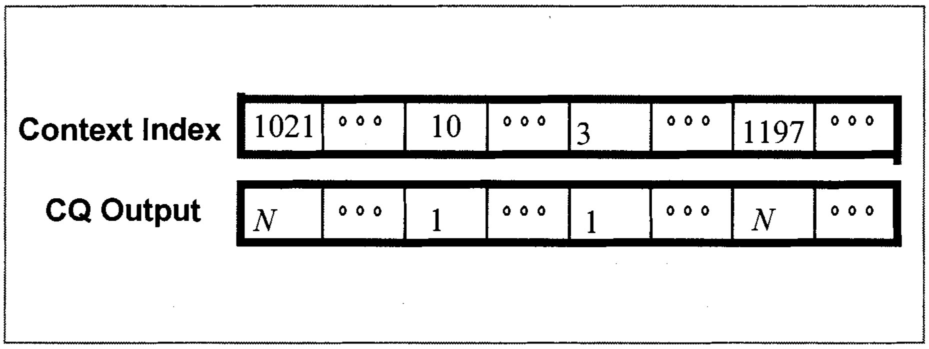
4) Stop if $m = P-1$

5) $m = m+1$, go to step #2.

Once the last boundary, b_p, has been tested, the context quantizers and the new context space are both determined. The size of the new context space will be much smaller than the original one. Therefore the side information to describe its partition will also be reduced.

The parameter ε is critical, since it controls how many boundaries will be kept. A large ε produces coarser context quantizers, which implies a more severe penalty to the data-stream rate. Although the side information in this case will be small, the total rate may be very high. On the other hand, if ε is too small, most boundaries will be kept, which will result in a large amount of side information. Therefore ε should be chosen carefully to minimize the total bit rate of data and side information.

3.2.2.2. Entropy Coded State Sequence (ECSS)



Context Index	1021	∘ ∘ ∘	10	∘ ∘ ∘	3	∘ ∘ ∘	1197	∘ ∘ ∘
CQ Output	N	∘ ∘ ∘	1	∘ ∘ ∘	1	∘ ∘ ∘	N	∘ ∘ ∘

Figure 3-14 Illustration of State Sequence

Another approach to reducing the side information is to decrease the amount of bits spent on the individual entries in the context book. In this section, we propose a method to achieve the above goal indirectly. The state sequence, which is a sequence of *M* context quantizer output indices, is sent as side information instead of sending the context book. The context index is much larger than its corresponding context quantizer output index. The *M* context quantizer output indices will be transmitted in some order such that the corresponding context indices can be calculated at both the encoder and the decoder. As a result, the context book can be built on the fly instead of being sent as side information.

We now need to determine the order in which to send the context quantizer output indices. Let's first look at the input sequence itself, $X_0, X_1, X_2, \ldots, X_Z$. The context index, $C(X_i)$, of each symbol X_i can be calculated after a causal context template is defined. When the input symbols are coded sequentially, different contexts occur in order. Of course, most of them appear more than once, that is, $C(X_i) = C(X_j)$ for some $i \neq j$. This order information can be exploited to reduce the bits for side information. Instead of transmitting the large context index $C(X_i)$, its context quantizer output index $Q(C(X_i))$ can be sent when $C(X_i)$ first appears. Both the encoder and the decoder will build a context book on the fly to memorize the classification information of the contexts that have already appeared. Therefore, when the contexts occur again, it is not necessary to transmit their corresponding quantizer outputs. At the end of coding the input sequence, this context book, built at both encoder and decoder, will be the same as the original one which is actually not transmitted.

The scheme is formally stated as the following.

1) The original context book is designed using the context quantization method proposed in Section 3.1.

2) Initialization: $i = 0$, the context book B which will be built on the fly is set to be empty. Start coding the input sequence from X_0.

3) For each X_i, the context $C(X_i)$ is calculated.

4) Check if $C(X_i)$ is already in the context book B. If yes, obtain its context quantizer output index and code X_i using arithmetic coder. Otherwise, add $C(X_i)$ to the subset $Q(C(X_i))$ in B. Send $Q(C(X_i))$.

5) Stop if $i = Z$, which is the end of sequence.

6) $i = i+1$, go to step #3.

In this scheme, the resulting side information turns out to be a sequence of M context quantizer output indices, which we call the state sequence. The number of entries in the state sequence is the same as in the context book, however the bits spent to represent entries in the state sequence is much less than the bits spent to directly represent the entries in the context book. Moreover, the state sequence can be further compressed using entropy coding by exploiting the fact that the populations of different subsets are unequal.

3.2.3. Coding Process

Based on a combination of the techniques in the previous sections, we summarize the proposed scheme into the following steps.

1) Use the coarse context quantization method to generate a reduced context space.

2) Obtain the optimal partition of the reduced context space using the context quantization algorithm proposed in Section 3.1.

3) Store the context mapping information in the form of a context book which won't be transmitted directly to the decoder.

4) Build the state sequence on the fly according to the original context book and send entropy coded state sequence as side information when sequentially coding the input sequence.

The resulting side information now consists of the description of the set of coarse context quantizers, plus the entropy coded state sequence. This is much more efficient than transmitting the original context book directly.

3.2.4. Experimental Results

We perform experiments on images processed by a wavelet transform to a depth of three, where the filter set is the standard 9-7 configuration [54]. We quantize the transformed data using uniform quantizers. Quantizers are designed for each subband by determining the difference between the maximum and minimum values of the coefficients and dividing this number by 16, the desired number of levels. This last number is set fairly arbitrarily since our focus is on the entropy coding. The context template is defined using the four causal nearest neighbors, resulting in a raw count of 65,536, most of which have empty histograms.

We first apply the context quantization scheme described in Section 3.1 to the subband images. If we take a direct approach to describing the context book, the amount of information needed to represent it will be very large. Table 3.3 shows the results for subband LH_2 of image Barbara. The data rate for the bit stream is 1.41bpp when the context space is quantized to N=4 subsets, but the side information needed to list the context mappings using a straight binary code for each context label is another 1.41bpp, and the total rate of the data and side information is 2.82bpp. When we encode it without context quantization, the rate is 1.69 bpp. Although it suffers from a severe context dilution problem, it is much better than the one applying the context quantization method due to the inefficiency of coding the side information. However, the bits spent on the side information can be significantly reduced when we transmit the entropy coded state sequence instead of the original context book. A minor drawback of the state sequence method is the increase in side information rate with N due to the increase in possible values for each entry in the state sequence. The bit rate for coding the state sequence of all of the subbands in Barbara using arithmetic coder when N =2 is given in Table 3.2. In comparison with the direct strategy of a binary code for each context label, the side information is drastically reduced by 90%-95%. Although the improvement for a larger N is not so dramatic, it is still strikingly shrunk.

The side information can be further compressed if the coarse context quantization method is used for constructing a small context space. However, in this case we need to pay the price of increasing data rate. This trade-off between the data rate and side information rate is controlled by the parameter ε, and we can adjust ε to reach the minimum total rate for data and side information. The results are shown in Table 3.4 to Table 3.8 for various subbands of three 512x512 images. As can be seen, the performance is substantially improved after applying the proposed methods. Considering LH_2 of Barbara specifically, the rate drops from 2.82 bpp to 1.56 bpp when only the entropy coded state sequence (ECSS) method is applied. Note the large drop in the number of bits spent on side information. When the coarse context quantization (CCQ) method is applied before context quantization, the side information rate decreases by another 0.12 bpp while the data rate increases by 0.05 bpp. Therefore, the overall performance is further improved.

Table 3.2 Side information bit rate (bpp) for subbands in Barb

Method	LL_0	HL_1	LH_1	HH_1	HL_2	LH_2	HH_2	HL_3	LH_3	HH_3
ECSS	0.52	0.20	0.18	0.09	0.07	0.08	0.16	0.05	0.05	0.01
Direct	8.60	3.15	2.97	1.38	1.72	1.41	2.50	0.9	0.81	0.25

Table 3.3 Results for coding subband *LH_2* (bpp) in Barbara N = 4, ε =0.02

Method		data rate	side-info rate	total rate
No Context Quantization		1.69	0	1.69
Context Quantization	Direct	1.41	1.41	2.82
	ECSS	1.41	0.15	1.56
	CCQ+ECSS	1.46	0.03	1.49

Table 3.4 Results for coding subband LH_3 (bpp) in Barb N =16, ε =0.01

Method		data rate	side-info rate	total rate
No Context Quantization		1.48	0	1.48
Context Quantization	Direct	1.23	0.81	2.04
	ECSS	1.23	0.18	1.41
	CCQ+ECSS	1.30	0.05	1.35

Table 3.5 Results for coding subband HH_1 (bpp) in Goldhill N = 2, ε =0.06

Method		data rate	side-info rate	total rate
No Context Quantization		2.86	0	2.86
Context Quantization	Direct	2.10	3.69	5.79
	ECSS	2.10	0.23	2.33
	CCQ+ECSS	2.18	0.006	2.186

Table 3.6 Results for coding subband HL_3 (bpp) in Goldhill N =8, ε =0.005

Method		data rate	side-info rate	total rate
No Context Quantization		1.43	0	1.43
Context Quantization	Direct	1.28	0.52	1.80
	ECSS	1.28	0.08	1.36
	CCQ+ECSS	1.29	0.03	1.32

Table 3.7 Results for coding subband LL_0 (bpp) in Baboon N = 8, ε =0.005

Method		data rate	side-info rate	total rate
No Context Quantization		3.62	0	3.62
Context Quantization	Direct	1.32	7.82	9.14
	ECSS	1.32	2.02	3.34
	CCQ+ECSS	1.29	2.02	3.31

Table 3.8 Results for coding subband HL_2 (bpp) in Baboon N =4, ε =0.0035

Method		data rate	side-info rate	total rate
No Context Quantization		2.59	0	2.59
Optimal Context Quantization	Direct	2.00	2.38	4.38
	ECSS	2.00	0.29	2.29
	CCQ+ECSS	2.15	0.01	2.16

3.2.5. Conclusions

In this section we present two techniques for efficiently describing the context book of the context quantizer.

CCQ is to decrease the number of the entries in the context book by automatically constructing a new small context space. In the previous section, we presented a locally optimal context quantizer design method to minimize the data rate. In this section, our goal is to minimize the sum of code length for both the side information and the input data. We allow sub-optimal groupings in the sense of minimizing the data rate in this scheme. Although these groupings will increase the data rate, the resulting context quantizer mappings are easier to describe and it will save the side information bits. Another feature of this scheme is that it can pre-process large context space to improve the efficiency of the context quantizer design algorithm described in Section 3.1. When high quality compression is required in some applications, high bit rate quantizers will be applied. In this case, a huge context space will be generated. The context quantizer design algorithm described in Section 3.1 will be very slow and inefficient. Pre-processing of the context space using CCQ scheme will produce a moderate size context space which can be handled more efficiently by the context quantizer design algorithm.

ECSS method is designed for reducing the bits spending on the individual entry in the context book. This goal is achieved indirectly. Instead of transmitting the context book, the entropy coded state sequence with small context quantizer output indices is sent as the side information. The context book can be built on the fly at both encoder and decoder according to the state sequence. Unlike CCQ, this scheme has no effect on the data rate. The only objective of this scheme is to minimize the side information bits. It will be applied to binary sources which we will discuss in the next chapter.

CHAPTER 4.
CONTEXT QUANTIZATION FOR ENTROPY CODING OF BINARY SOURCE

4.1. Motivation

In the previous chapter we proposed a context quantizer design algorithm for non binary sources. The algorithm is essentially a vector quantization (VQ) approach that clusters raw context instances using Kullback-Leibler distance as the VQ distortion metric. The context quantizer design is done by a variant of the generalized Lloyd method of gradient descent, and consequently the solution is only locally optimal.

In this chapter, we work with binary sources. A non-binary source can be converted to a binary source by a sequence of binary decisions and coded as the binary sequence. If the source data is binary, then the probability simplex space is one dimensional. This reduces context quantizer design to a scalar quantizer design problem, and consequently the problem can be solved by dynamic programming and the solution can be made globally optimal.

An important operational issue in context quantization, which has not been satisfactorially solved, is how to compactly describe the inverse quantizer mapping function to the decoder. We consider two approaches for this. The first one is two-pass, involving designing the optimal context quantizer based on the count statistics of the input data and sending the quantizer mapping function directly using a huge look-up table as side information. The length of this side information is usually significant, canceling compression gain made by the optimal context quantizer. The second approach is to optimize the context quantizer with respect to the statistics of a training set. In this case, the context quantizer is fixed and known to both encoder and decoder. There is no need to transmit any side information. However, an ensuing question is how to handle any mismatch in statistics between the training set and the input. This problem, which has remained largely untreated, is the main concern of this chapter. Intuitively, the use of training set in context quantizer design can speed up the adaptation process of an arithmetic coder that learns from some suitable preknowledge rather than from scratch. Unless the statistics of the training set and the input source match perfectly, there exists an optimal blend of the two statistics to achieve the minimum adaptive code length.

4.2. Minimum Conditional Entropy Context Quantization – Binary Case

In this section, we first briefly review the work of minimum conditional entropy context quantization (MCECQ) for binary case [39].

Let X be a discrete random variable, and let C be a jointly distributed random vector, possibly real. Given a positive integer M, we wish to find the quantizer $Q: C \to \{1,2,...,M\}$ such that $H(X \mid Q(C))$ is minimized. Clearly, $H(X \mid C) > H(X \mid Q(C))$ by the convexity of H. However, we wish to make

$H(X \mid Q(C))$ as close to $H(X \mid C)$ as possible. Equivalently, we wish to minimize the non-negative "distortion" of Q

$$D(Q) = H(X \mid Q(C)) - H(X \mid C) \qquad (4.1)$$

The quantized regions $A_m = \{\mathbf{c} : Q(\mathbf{c}) = m\}, m = 1,\ldots,M,$ of an (optimal) minimum conditional entropy context quantizer are generally quite complex in shape, and may not even be convex or connected. However, their associated sets of pmfs $B_m = \{P_{X|C}(\cdot \mid \mathbf{c}) : \mathbf{c} \in A_m\}$ are simple convex sets in the probability simplex for X, owing to the necessary condition for optimal Q.

If X is a binary random variable, then its probability simplex is one-dimensional. In this case, the quantization regions B_m are simple intervals. If the random variable Z is defined as $P_{X|C}(1 \mid C)$ (the posterior probability that $X = 1$ as a function of C), then the conditional entropy $H(X \mid Q(C))$ of the optimal context quantizer can be expressed by

$$H(X \mid Q(C)) = \sum_{m=1}^{M} P\{Z \in [q_{m-1}, q_m)\} H(X \mid Z \in [q_{m-1}, q_m)) \qquad (4.2)$$

for some set of thresholds $\{q_m\}$ specifying the quantization regions B_m. Therefore the minimum conditional entropy context quantizer (MCECQ) can be found by searching over $\{q_m\}$. This is a scalar quantization problem, which can be solved exactly using dynamic programming [64]. this way, the problem of optimal MCECQ design is reduced to one of scalar quantization, regardless of the dimensionality of the context space. Once the scalar problem is solved, the optimal MCECQ cells A_m are given by

$$A_m = \{\mathbf{c} : P_{X|C}(1 \mid \mathbf{c}) \in [q_{m-1}, q_m)\} \tag{4.3}$$

4.3. Structure and Complexity of Context Quantizer Mapping

Unfortunately, the optimal partition of the context space by $A_m, m = 1,2,...,M$ is highly complex [39]. Now we see that the boundary between any two adjacent MCECQ cells consists of vectors $\mathbf{c}$ for which the posterior probability $P_{X|C}(1 \mid C)$ is a constant. Specifically, it follows that $P_{X|C}(1 \mid C) = q_m$ for $\mathbf{c}$ along the boundary between A_m and A_{m+1} . Equivalently, A_m can be expressed in terms of the likelihood ratio.

$$L(\mathbf{c}) = \frac{P_{C|X}(\mathbf{c}|1)}{P_{C|X}(\mathbf{c}|0)} = \frac{P_X(0)}{P_X(1)} \frac{P_{X|C}(1 \mid \mathbf{c})}{1 - P_{X|C}(1 \mid \mathbf{c})}, \tag{4.4}$$

which is a strictly increasing function f of the posterior probability $P_{X|C}(1 \mid C)$, as

$$A_m = \{\mathbf{c} : L(\mathbf{c}) \in [f(q_{m-1}), f(q_m))\} \tag{4.5}$$

Hence the likelihood ratio $L(\mathbf{c})$ along the boundary between A_m and A_{m+1} is a constant.

The constant likelihood ratio on the boundary of MCECQ cells is a useful property to study the geometry of MCECQ in the context space of $\mathbf{c}$. Now assume that the conditional densities $P(C \mid X = 0)$ and $P(C \mid X = 1)$ belong to the family of Kotz-type *d*-dimensional elliptical distributions in which the density functions take the form

$$f_{C|X}(\mathbf{c}\mid 0)=\alpha_d(r_0,s_0)\mid\Sigma_0\mid^{-1/2}\exp\{-r_0[(\mathbf{c}-\mu_0)'\,\Sigma_0^{-1}(\mathbf{c}-\mu_0)]^{s_0}\}$$

$$f_{C|X}(\mathbf{c}\mid 1)=\alpha_d(r_1,s_1)\mid\Sigma_1\mid^{-1/2}\exp\{-r_1[(\mathbf{c}-\mu_1)'\,\Sigma_1^{-1}(\mathbf{c}-\mu_1)]^{s_1}\} \qquad \textbf{(4.6)}$$

This family of joint distributions includes the Gaussian distribution as a special case. The likelihood ration of (4.6) is given by

$$L(\mathbf{c})=\frac{f_{C|X}(\mathbf{c}|1)}{f_{C|X}(\mathbf{c}|0)}= \qquad \textbf{(4.7)}$$
$$\alpha(d,r_0,r_1,s_0,s_1)\exp\{r_0[(\mathbf{c}-\mu_0)'\,\Sigma_0^{-1}(\mathbf{c}-\mu_0)]^{s_0}-r_1[(\mathbf{c}-\mu_1)'\,\Sigma_1^{-1}(\mathbf{c}-\mu_1)]^{s_1}\}$$

where $\alpha(d,r_0,r_1,s_0,s_1)$ is a constant independent of $\mathbf{c}$. Since $L(\mathbf{c})$ is a constant on the boundary of MCECQ quantizer cell, it follows from that the boundary points $\mathbf{c}$ satisfy

$$r_0[(\mathbf{c}-\mu_0)'\,\Sigma_0^{-1}(\mathbf{c}-\mu_0)]^{s_0}--r_1[(\mathbf{c}-\mu_1)'\,\Sigma_1^{-1}(\mathbf{c}-\mu_1)]^{s_1}+\beta(d,r_0,r_1,s_0,s_1)=0 \qquad \textbf{(4.8)}$$

where $\beta(d,r_0,r_1,s_0,s_1)$ is another constant independent of $\mathbf{c}$. Thus the MCECQ cells are sets bounded by polynomial surfaces. In particular, if both $f_{C|X}(\mathbf{c}\mid 0)$ and $f_{C|X}(\mathbf{c}\mid 1)$ are *d*-dimensional Gaussians, a special case of Kotz-type elliptical distribution family with $s_0=1$ and $s_1=1$, then MCECQ cells are bounded by *d*-dimensional quadratic surfaces as immediately from (4.8). For the above quite large class of joint distributions, the context quantization function $Q(\bullet)$ can be simply defined as a parametric classifier that maps a point $\mathbf{c}$ in context space into a coding state $Q(\mathbf{c})$.

In Figure 4.1 and Figure 4.2 we plot MCECQ cells for two different two-dimensional Guassian distributions. Figure 4.1 presents the general arrangement of $P_{C|X}(\bullet\,|\,0)$ and $P_{C|X}(\bullet\,|\,1)$, and the corresponding MCECQ of three cells. In Figure 4.2, we give a special case and also a worst case of MCECQ in terms of improving coding efficiency via context-based coding. In this example $P_{C|X}(\bullet\,|\,0)$ and $P_{C|X}(\bullet\,|\,1)$ have identical means, and the two underlying clusters are hence least separable from eachother. Nevertheless, as long as the two distributions $P_{C|X}(\bullet\,|\,0)$ and $P_{C|X}(\bullet\,|\,1)$ are not identical, then MCECQ can realize some coding gain over non-conditional entropy coding i.e., $H(X\,|\,Q(C)) \leq H(X)$. For instance, in the situation depicted by Figure 4.2 we have $H(X\,|\,Q(C)) = 0.8$ versus $H(X) = 1$.

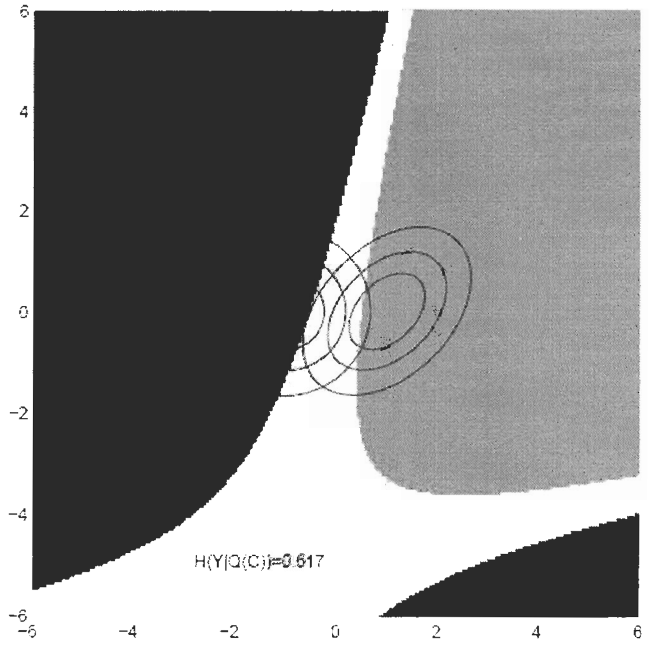


Figure 4-1 Two 2-dimensional Gaussian distributions, and the corresponding MCECQ 3-partition of the context space

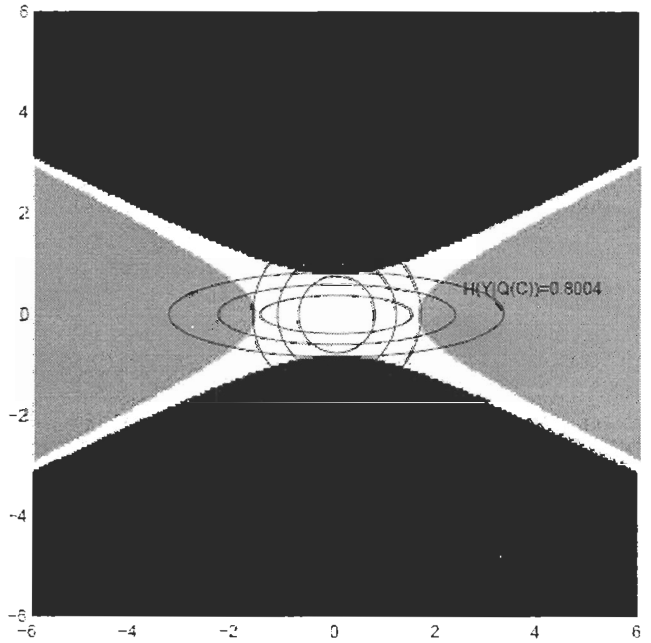


Figure 4-2 Two overlapped Gaussian distribution of different covariance matrices, and the corresponding MCECQ 3-partition of the context space.

4.4. Minimum Description Length Context Quantizer Design

4.4.1. Introduction

In the discussion above, we did not take the bits required to describe the context quantizer cells into account. In this case the MCECQ is only useful as a procedure to establish a lower bound on the achievable code length. A real challenge in applying an optimal context quantizer to data compression is how to describe the quantizer mapping $Q(\mathbf{c})$ with little or no side information, and at a reasonable computational complexity. For general $P(C \mid 0)$ and $P(C \mid 1)$, the quantization cells of MCECQ in context space have a very complex geometry and topology. Only for $P(C \mid 0)$ and $P(C \mid 1)$ that are Gaussian or some variants of Gaussian (Kotz-type joint distributions) do we have a tractable analytical description for $Q(\bullet)$.

The simplest way of implementing an arbitrary quantizer mapping $Q(\bullet)$ is to use a look-up table. But since $|C|$, the number of all possible raw contexts, is very large for high-order contexts, building a huge table of $|C|$ entries for $Q(c)$ is clearly impractical. If we make this table image-dependent, the compression gain will be cancelled by the high cost of sending large side information. Of course, the training set can be used to design the MCECQ and then fix the context quantizer in the actual coding. In this case, there is no need of sending side information since both encoder and decoder use the same fixed context quantizer. However, how to deal with the rare context instance, which are absent in the training set but present in the input data to be coded, becomes an issue. A simple way is to lump together all the rare context instances into a single coding state, which is clearly suboptimal. In this chapter we propose a method to send this information as side information and design the optimal context quantizer to minimize the description length which is the sum of the code length of data and side information.

4.4.2. Context Quantizer Design

4.4.2.1. Optimization of Context Quantization

When the input source processes a novel context instance that is not in the training set, which we call "rare" context instance, the encoder needs to signal the event to the decoder and identifies the quantized conditioning state in which arithmetic coding is performed. The required side information should be factored into the adaptive code length as well. Another design parameter to be optimized is the number M of context quantizer cells. The value of M not only regulates the impact of context dilution but also affects the length of side information to describe the quantizer mapping function to the decoder. The main contribution of our work is a unified treatment of all the above design parameters in the principle of minimum description length. This allows us to develop an optimal context quantizer design algorithm for minimum sum of adaptive arithmetic code length and the side information length, given the training set and the input source.

Since the population of the "rare" context instances is relatively small compared with the total number of distinct context instances which appear in the source data, one can send coded quantizer indexes (labels of coding states for the entropy coder) of the rare context instances. Associated with the input sequence $X_0, X_1, X_2, \ldots$ is the sequence of raw context instances $\{\mathbf{c}(X_i)\}$, with respect to a given context template. The first occurrences of different "rare" context instances are in the same order as the sequential coding of the input symbols. Many of the rare context instances appear more than once, that is, $\mathbf{c}(X_i)= \mathbf{c}(X_j)$ for some $i \neq j$. But the encoder only needs to code the first occurrence of a quantizer output index $Q(\mathbf{c}(X_i))$. Since the decoder observes the same source sequence as the encoder, both the encoder and the decoder can build a dictionary of the already appeared rare context instances on the fly. Therefore, when a rare context instance occurs again, the decoder knows its quantized value, i.e., the coding state for the entropy decoder.

In this scheme, the resulting side information to code the quantizer mapping function Q turns out to be a sequence Θ of n coding state indices, where n is the number of distinct rare context instances in the input sequence, which we call the state sequence. The state sequence Θ can be compressed by entropy coding because the distribution of different coding states is non-uniform.

Given the proposed side information coding scheme, we can formulate the problem of adaptive context quantization, in the principle of minimum description length (MDL), as one of minimizing the sum of the code length emitted by the adaptive arithmetic coding whose conditioning states are the MCECQ cells and the length of side information.

Let $Z = P_{X|C}(1 \mid \mathbf{c})$, $\underline{z} = \min Z$, $\bar{z} = \max Z$, and denote $Q(\tau, m)$ the set of all possible m-dimensional vectors $\mathbf{q} = (q_1, q_2, \ldots, q_m)$ such that

$$\underline{z} \equiv q_0 < q_1 < q_2 < \cdots < q_{m-1} < q_m = \tau < \bar{z} \tag{4.9}$$

Then the optimal context quantizer that minimizes the description length is given by

$$\hat{\mathbf{q}} = \arg \min_{\mathbf{q} \in Q(z,M)} \sum_{m=1}^{M} N\{Z \in (q_{m-1}, q_m]\} H(X \mid Z \in (q_{m-1}, q_m]) + n\{Z \in (q_{m-1}, q_m]\} \log \frac{n\{Z \in (q_{m-1}, q_m]\}}{n\{Z \in [\underline{z}, \bar{z}]\}} \tag{4.10}$$

where $N\{Z \in (q_{m-1}, q_m]\}$ is the number of the samples whose conditional probability $Z = P_{X|C}(1 \mid \mathbf{c})$ fall into the interval $(q_{m-1}, q_m]$ and $n\{Z \in (q_{m-1}, q_m]$ is the number of the distinct "rare" contexts whose $Z = P_{X|C}(1 \mid \mathbf{c})$ fall into the interval $(q_{m-1}, q_m]$. Clearly, in (4.10) the first term represents the code length of the conditional entropy coding based on the resulting optimal context quantizer and the second term is the length of the side information.

Let

$$\begin{aligned} L_0(q_{m-1}, q_m] &= N\{Z \in (q_{m-1}, q_m]\} H(X \mid Z \in (q_{m-1}, q_m]) \\ L_1(q_{m-1}, q_m] &= n\{Z \in (q_{m-1}, q_m]\} \log \frac{n\{Z \in (q_{m-1}, q_m]\}}{n\{Z \in [\underline{z}, \bar{z}]\}} \end{aligned} \tag{4.11}$$

(4.10) can be simplified to be

$$\overset{\wedge}{\mathbf{q}} = \arg \min_{\mathbf{q} \in Q(z,M)} \sum_{m=1}^{M} L_0(q_{m-1}, q_m] + L_1(q_{m-1}, q_m] \quad (4.12)$$

The optimal *M*-level context quantizer $\overset{\wedge}{\mathbf{q}}$ as given by (4.12) cam be efficiently computed by observing the following recursion

$$\begin{aligned} \overset{\wedge}{\mathbf{q}} &= \arg \min_{\mathbf{q} \in Q(r,j)} \sum_{m=1}^{j} L_0(q_{m-1}, q_m] + L_1(q_{m-1}, q_m] = \\ &\min_{\tau < r} \left\{ \min_{\mathbf{q} \in Q(r,j-1)} \sum_{m=1}^{j-1} L_0(q_{m-1}, q_m] + L_1(q_{m-1}, q_m] + L_0(\tau, r] + L_1(\tau, r] \right\} \end{aligned} \quad (4.13)$$

The recursion means that the solution for the problem of size *j* can be constructed from the solution of subproblems of size *j*-1. Because of this property that is called the principle of optimality in the optimization literature, we can use a straightforward dynamic programming [55] algorithm to solve (4.12)

Only the resulting optimal context quantizer thresholds $\{q_m\}$ and the entropy-coded state sequence Θ need to be sent as the side information. This suffices to specify the context quantizer mapping function Q to the decoder. Note that the resulting thresholds $\{q_m\}$ are image-dependent, which is different in general from the context quantizer optimized for the training set. Therefore, the context quantizer is optimized in the MDL sense for the input image, and made robust even if the input image and the training set have different statistics.

4.4.2.2. Implementation of Context Quantization

In order to design MDL-based optimal context quantizer, we must have an estimate of $P(X \mid C)$. In practice, $P(X \mid C)$ is seldom known exactly. Otherwise one would let an entropy coder directly operate on $P(X \mid C)$. In data compression applications $P(X \mid C)$ is either estimated from a suitable set of training data, or from the input data. A common estimate of $P_{X|C}(1 \mid \mathbf{c})$ is given by

$$\hat{P}_{X|C}(1 \mid \mathbf{c}) = \frac{n_1 + \delta}{n_0 + n_1 + 2\delta} \tag{4.14}$$

where n_0 and n_1 are the number of occurrences of 0 and 1, respectively, in a given context $\mathbf{c}$, and $\delta \in [0,1]$ is a parameter of the estimator.

Replacing the probability values with estimates thereof MDL-based context quantizer design becomes solving the optimization problem

$$\min_{q,M} \sum_{m=1}^{M} N\{\hat{P}_{X|C}(1 \mid \mathbf{c}) \in (q_{m-1}, q_m]\} H(X \mid \hat{P}_{X|C}(1 \mid \mathbf{c}) \in (q_{m-1}, q_m])$$

$$+ n\{\hat{P}_{X|C}(1 \mid \mathbf{c}) \in (q_{m-1}, q_m]\} \log \frac{n\{\hat{P}_{X|C}(1 \mid \mathbf{c}) \in (q_{m-1}, q_m]\}}{n\{\hat{P}_{X|C}(1 \mid \mathbf{c}) \in [0,1]\}} \tag{4.15}$$

We can only claim that dynamic programming algorithm can design the context quantizer minimizing the description length, under the constraint that the quantizer cells are contiguous intervals on the values of $\hat{P}_{X|C}(1 \mid \mathbf{c})$. It remains an open problem whether the constraint leads to the overall minima.

We use both the statistics of a suitable training set and the sample statistics of the input image to estimate $\hat{P}_{X|C}(1\,|\,\mathbf{c})$. We refer to those context instances that are observed in the training set as "common", and those that only appear in the input image as "rare". For the common context instances **c** we use the conditional probability $\hat{P}_{X|C}(1\,|\,\mathbf{c})$ estimated from the training set. On the other hand, for the rare context instances $\hat{P}_{X|C}(1\,|\,\mathbf{c})$ are estimated using the sample statistics of the input image. These estimated conditional probabilities $\hat{P}_{X|C}(1\,|\,\mathbf{c})$ are sorted in ascending order in the probability simplex space, regardless whether **c** is rare or common. This linear ordering enables an optimization approach of dynamic programming described in 4.4.2.1 to design *M*-level context quantizer. This *M*-level context quantizer can be globally optimized for minimum description length, which is better than a gradient descent method that can get trapped in a local minimum.

If we normalize the size of the training set to be the length of an input sequence, then the dynamic programming algorithm can automatically decide the optimal number *M* of coding contexts for the input size. This is simply done by increasing the number of context quantizer cells one at a time in the bottom-up dynamic programming process, until reaching the point when the actual code length starts to increase due to context dilution.

4.4.3. Experimental Results

We conducted experiments on lossless coding of binary source, halftone images, which are among the most difficult to compress. Thus they present great challenges to context based entropy coding. Consequently, they serve as good, demanding test cases for the performance of different context quantizers.

Halftoning or analog halftoning is a process that simulates shades of gray by varying the size of tiny black dots arranged in a regular pattern. This technique is used in printers, as well as the publishing industry. If you inspect a photograph in a newspaper, you will notice that the picture is composed of black spots even though it appears to be composed of greys. This is possible because of the spatial integration performed by our eyes. Our eyes blend fine details and record the overall intensity. Digital halftoning is similar to halftoning in which an image is decomposed into a grid of halftone cells. Elements of an image are simulated by filling the appropriate halftone cells. The more number of black dots in a halftone cell, the darker the cell appears. For example, in Figure 4.3 a tiny dot located at the center is simulated in digital halftoning by filling the center halftone cell; likewise, a medium size dot located at the top-left corner is simulated by filling the four cells at the top-left corner. The large dot covering most of the area in the third image is simulated by filling all halftone cells.

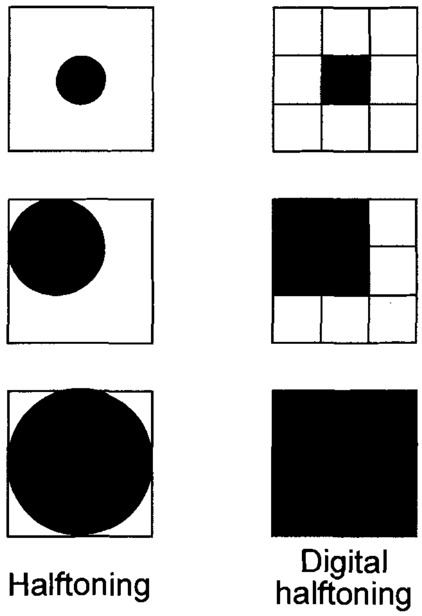
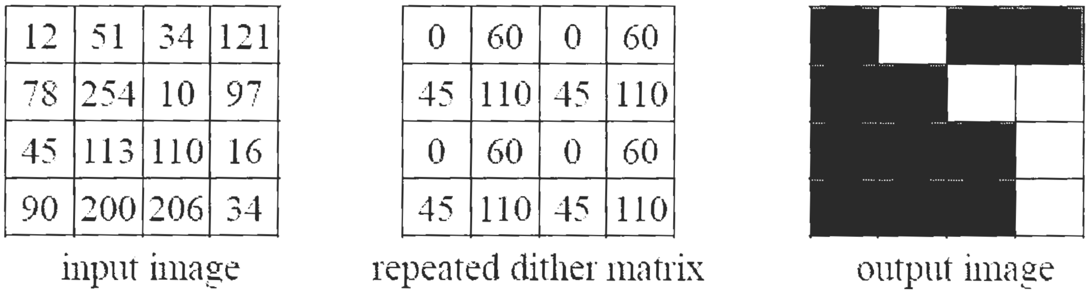


Figure 4-3 Digital halftoning

Two methods for generating digital halftoning images are used in our experiments, dithering and error diffusion.

Dithering technique creates an output image with the same number of dots as the number of pixels in the source image. Dithering can be thought of as thresholding the source image with a dither matrix. The matrix is laid repeatedly over the source image. Wherever the pixel value of the image is greater than the value in the matrix, a dot on the output image is filled. Figure 4-4 shows a sample of the dithering operation. Figure 4-5 shows a sample of dithering halftone image that we use in our experiments.



12	51	34	121
78	254	10	97
45	113	110	16
90	200	206	34

input image

0	60	0	60
45	110	45	110
0	60	0	60
45	110	45	110

repeated dither matrix

■	□	■	■
■	■	□	□
■	■	■	□
■	■	■	□

output image

Figure 4-4 Dithering Operation

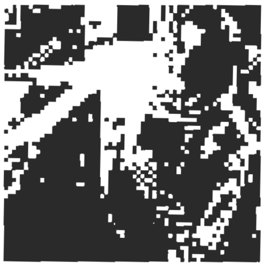


Figure 4-5 Sample of dithering halftone image

Error diffusion is also called spatial dithering. It sequentially traverses each pixel of the source image. Each pixel is compared to a threshold. If the pixel value is higher that the threshold, a 255 is outputted; otherwise, a 0 is outputted. The error, which is the difference between the input pixel value and the output value, is dispersed to nearby neighbors. Error diffusion is a neighborhood operation since it operates not only on the input pixel, but also its neighbors. Generally, neighborhood operations produce higher quality results than point operations. Error diffusion, when compared to diterhing, does not generate those artifact introduced by fix thresholding matrices. However it is more challenging to the context based entropy coding schemes. Figure 4-6 shows an example of error diffusion halftone images which we used for tests.

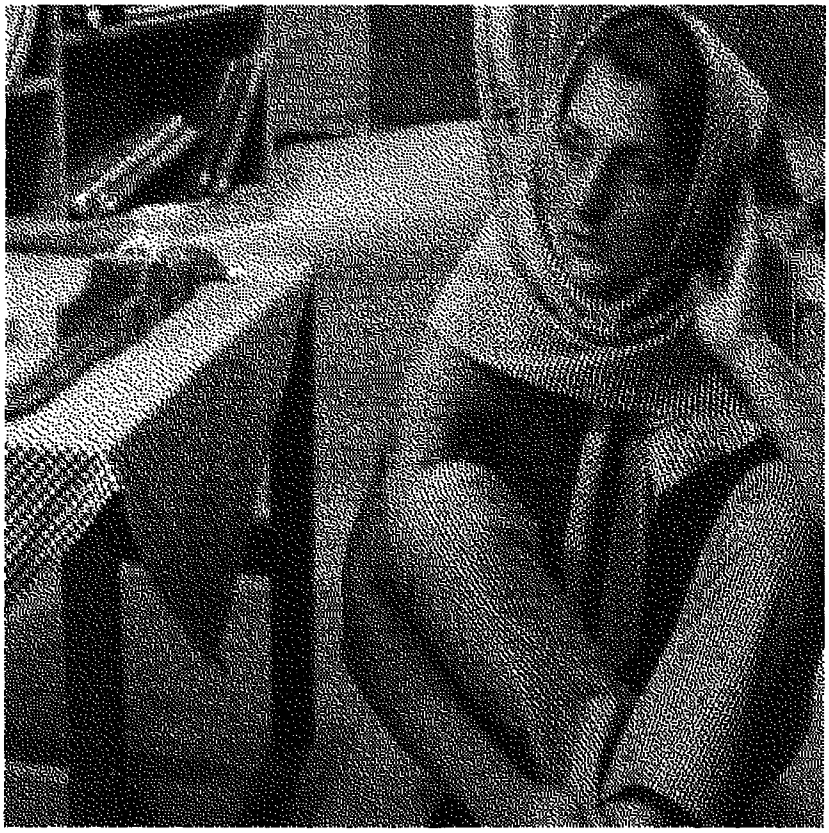


Figure 4-6 Sample of error diffusion halftone image

We implemente the proposed MDL-based context quantizers and evaluated them in lossless coding of dithering and error diffusion halftone images. The training set of raw contexts is generated out of 13 halftone images that were converted from benchmark grayscale images on the Internet. The test set consisting of images *Barbara, Lena,* and *Mandrill*, is disjoint from the training set.

In order to evaluate the performance of our MDL-based context quantizer scheme, we compare it with JBIG2 standard. Here we applied the four context templates defined in JBIG2 on the test images and chose the one with the best performance. Table 4.1 and Table 4.2 show the bit rates obtained by the MDL-based context quantizers and the adaptive entropy coders defined in JBIG2. Our scheme outperforms JBIG2 on the two sets of halftone images by 16% and 4% on average respectively. As the results show, error diffusion halftone images are more difficult to compress than the dithering halftone images.

Table 4.1 Bit rates of dithering halftone images

Image	***MDLCQ***	***JBIG2***
Barbara	0.54	0.65
Lena	0.40	0.48
Mandrill	0.71	0.84

Table 4.2 Bit rates of error diffusion halftone images

Image	***MDLCQ***	***JBIG2***
Barbara	0.722	0.770
Lena	0.686	0.744
Mandrill	0.806	0.841

4.5. Image Dependent Context Quantizer Design with Efficient Side Information

MDL-based context quantizer described in the previous section is still designed mainly based on the training set statistics. The estimated conditional probabilities $\hat{P}_{X|C}(1\,|\,\mathbf{c})$ are used not only to design the optimal context quantizer but also to determine the context quantizer output $Q(\mathbf{c})$ for each context instance $\mathbf{c}$. $Q(\mathbf{c})$ is fixed during the actual coding. For common context instances, $\hat{P}_{X|C}(1\,|\,\mathbf{c})$ are estimated from the training set, while rare ones are from the input. However, rare context instances are only a small percentage of the total context instances. As a result, the resulting optimal context quantizer, defined by a set of thresholds$\{q_m\}$, is mainly determined by the training set statistics.

If there is any mismatch in statistics between the input and the training set, the optimality of the predesigned context quantizer can be compromised. One remedy is to employ an on-line algorithm that redesigns MCECQ for each X_i based on past samples$X_1, X_2, \cdots, X_{i-1}$. Although it can be done in theory, on-line MCECQ design is computationally too expensive to be practical. An alternative solution (sub-optimal) is to fix a predesigned context quantizer$Q$, but update $\hat{P}_{x|c}(1\,|\,\mathbf{c})$ on the fly as in adaptive arithmetic coding.

In this section we propose a scheme to design an image dependent context quantizer that can be described with very small amount of side information. This scheme is two-pass. In the first pass, the optimal context quantizer is designed based on the conditional probability estimated from the input instead of the training set. Only the centroids of the optimal context quantizer are sent as the side information. In the second pass, when the input is being coded, the conditional probability of each context instance is initialized as the estimate from the training set and then adaptively updated on the fly. These estimated conditional probabilities are used to determine the context quantizer output. Given the centroids, nearest neighbor search is applied. As a result, the context quantizer output of a specific context instance may change according to the more accurate estimate of the conditional probability along the way. This will improve the overall compression efficiency.

4.5.1. Context Quantizer Design

The proposed scheme is two-pass. During the first pass, the input image is raster scanned and the statistics, including conditional probabilities $Z = P_{x|\mathbf{c}}(1 \mid \mathbf{c})$ and the number of occurrence $N(\mathbf{c})$ of each context instance $\mathbf{c}$, which appear in the input image, are collected to design MCECQ using dynamic programming to minimize

$$\begin{aligned} &\min_{\mathrm{q},M} \sum_{m=1}^{M} N\{Z \in (q_{m-1}, q_m]\} H(X \mid Z \in (q_{m-1}, q_m]) \\ &= \min_{\mathrm{q},M} \sum_{m=1}^{M} N_m H(X \mid Z \in (q_{m-1}, q_m]) \end{aligned} \tag{4.16}$$

where N_m is the number of the samples whose conditional probability $Z = P_{x|\mathbf{c}}(1 \mid \mathbf{c})$ fall into the interval $(q_{m-1}, q_m]$. Let $\beta_m = P_{x|Q(\mathbf{c})}(1 \mid m)$ be the mth reproduction pmf. These reproduction pmfs are also the centroids of the resulting

MCECQ. It is easy to see that $\beta_m = \frac{n\{Z \in (q_{m-1}, q_m]\}}{N\{Z \in (q_{m-1}, q_m]\}} = \frac{n_m}{N_m}$, where n_m is the number of occurrences of 1 when $Z = P_{x|\mathbf{c}}(1 \mid \mathbf{c})$ fall into the interval $(q_{m-1}, q_m]$. The reproduction pmfs are sent as the side information and known by both encoder and decoder.

4.5.2. Estimation of Conditional Probabilities

For each context instance $\mathbf{c}$, we estimate its conditional probability $\hat{P}_{x|\mathbf{c}}(1 \mid \mathbf{c})$ first from a suitable training set, which is done off-line. In our experiment, a 16-order context template is used to code the input image. The default pixel ordering by 2-norm is shown in Figure 4-7

				14				
		12	10	6	9	11		
	16	8	4	2	3	7	15	
	13	5	1	X				

Figure 4-7 Default ordering of the past with maximum template size of 16

Let c^k be a *k*-order context. $P_{x|\mathbf{c}}(1 \mid c^k)$ denotes the conditional probability in the *k*-order context template. For the common context instances, which occur in the training set in the given 16-order context template, we estimate the conditional probability $\hat{P}_{x|\mathbf{c}}(1 \mid \mathbf{c})$ by $P_{x|\mathbf{c}}(1 \mid c^{16})$, which is obtained from the training set. On the other hand, for the rare context instances, which do not happen in the training set, we shrink the context template. That means

$\hat{P}_{x|c}(1\,|\,\mathbf{c}) = P_{x|c}(1\,|\,c^k)$ $(k < 16)$ are used as the conditional probability estimate. Each time we decrease size *k* by 1 according to the context pixel ordering shown in Figure 4-7until c^k $(k < 16)$ occur in the training set.

Note that the probability estimated from the training set is only used as an initial one and being continually updated during the process of coding the input image, which happens in the second pass. And the nearest neighbor search method is applied to determine the context quantizer output $Q(\mathbf{c})$ for each context instance **c**.

$$Q(\mathbf{c}) = \left\{i:\ \left|\hat{P}_{x|c}(1\,|\,\mathbf{c}) - \beta_i\right| \le \left|\hat{P}_{x|c}(1\,|\,\mathbf{c}) - \beta_j\right|,\ for\ all\ j \ne i,\ 1 \le i,j \le M\right\} \tag{4.17}$$

4.5.3. Coding Process

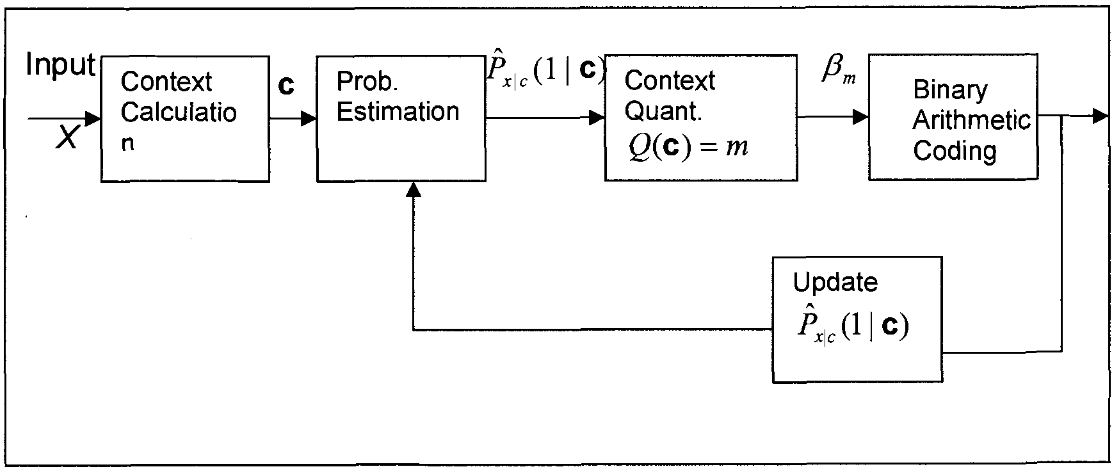


Figure 4-8 Diagram of the coding process

1) We use the method stated in section 4.5.1 to design the context quantizer,

whose centroids are $\beta_m = \frac{n_m}{N_m}$.

2) The initial conditional probability estimates $\hat{P}_{x|\mathbf{c}}(1\,|\,\mathbf{c})$ are collected from the scheme described in section 4.5.2.

3) Coding: for each input binary symbol X, determine the context instance $\mathbf{c}$ in the 16-order context template. Then $\hat{P}_{x|c}(1\,|\,\mathbf{c})$ is used to determine the context quantizer output $Q(\mathbf{c}) = m$ when $\left|\hat{P}_{x|c}(1\,|\,\mathbf{c}) - \beta_m\right| \le \left|\hat{P}_{x|c}(1\,|\,\mathbf{c}) - \beta_k\right|,\ \mathit{for\ all}\ m \ne k,\ 1 \le m,k \le M$. The number of bits spent on coding this symbol X is close to $\log_2 \beta_m$.

4) Update conditional probability estimates $\hat{P}_{x|c}(1\,|\,\mathbf{c})$ for the context instance $\mathbf{c}$

5) Repeat step2 and step3 until the whole image is coded.

4.5.4. Experimental Results

We tested the proposed image dependent context quantizer with little side information scheme on halftone images, a binary source which is very difficult to compress generated by error diffusion method. The training set and the test set are the same as described in section 4.4.3. In order to evaluate the performance of the scheme, we compare it with MDL-based method, JBIG and JBIG2 standard. The results are shown in Table 4.3. As you can see, the image dependent context quantizer design method is better than MDL-based method and it outperforms JBIG by 13% to 23% and JBIG2 by 4% to 8% for the chosen test images.

Table 4.3 Bit rate comparison between Image Dependent CQ and other schemes

Image	*Image Dependent CQ*	*MDLCQ*	*JBIG2*	*JBIG*
Barbara	0.713	0.722	0.770	0.867
Lena	0.674	0.686	0.744	0.884
Mandrill	0.800	0.806	0.841	0.927

4.6. Context Quantization for Minimum Adaptive Code Length

All context quantizers discussed up to now are designed for static arithmetic coding without adaptation. In other words, the arithmetic code is driven by a fixed probability estimate $\hat{P}(1 \mid \mathrm{Q}(\mathbf{c}))$. Static arithmetic coding is suboptimal when the source statistics is not stationary. Our next quest is to design the context quantizer that can minimize the actual code length of adaptive arithmetic coding.

A problem with the context quantizer described in the previous section is that only the input statistics is used to design the MCECQ in the first pass. Since the training set statistics may differ from the statistics of the input, we should minimize the effect of the mismatch between the training set and the input by adaptive arithmetic coding.

In this section, we are going to improve the previous context quantizer designs in the above two regards.

4.6.1. Context Quantizer Design

Now consider adaptive context-based arithmetic coding of a binary sequence x^I sequentially, using the probability estimate defined in (4.13) in each context instance. The probability estimates are updated on the fly for each of the I binary input symbols. Given a binary sequence x^I, its total code length $L(x^I \mid Q(\mathbf{c}))$ by adaptive context based arithmetic coding may be computed based on the set of counts (n_0, n_1) for all contexts without actually coding x^I. This is because the order of 0 and 1 appearance does not change the adaptive code length [40]. Let l_m be the adaptive code length of all symbols whose contexts fall into context quantizer cell $(q_{m-1}, q_m]$. Then we have [56]

$$l_m = \begin{cases} 0 & n_0 = 0 \; and \; n_1 = 0, \\ -\log \dfrac{\prod_{j=0}^{n_0-1}(\delta + j)\prod_{j=0}^{n_1-1}(\delta + j)}{\prod_{j=0}^{n_0+n_1-1}(2\delta + j)} & n_0 \neq 0 \; and \; n_1 \neq 0, \\ -\log \dfrac{\prod_{j=0}^{n_0+n_1-1}(\delta + j)}{\prod_{j=0}^{n_0+n_1-1}(2\delta + j)} & n_0 \neq 0 \; xor \; n_1 \neq 0, \end{cases} \tag{4.18}$$

Operationally, it is easy to apply the dynamic programming algorithm to the adaptive code length when designing the context quantizer The algorithm produces a context quantizer Q^* that can minimize the total adaptive code length, namely,

$$L(x^I \mid Q^*(\mathbf{c})) = \min_{M,Q(\mathbf{c})} L(x^I \mid Q(\mathbf{c})) = \min_{M,Q(\mathbf{c})} \sum_{m=1}^{M} l_m \tag{4.19}$$

As we stated above, our goal is to design an optimal context quantizer not only to minimize the adaptive code length, but also to minimize the effect of mismatch between the training set and the input. Taking these two elements into account, the objective function defined in (4.15) becomes

$$\min_{M,Q(\mathbf{c})} \sum_{m=1}^{M} \left\{ l_m(\alpha n_0' + n_0, \alpha n_1' + n_1) - l_m(\alpha n_0', \alpha n_1') \right\} \tag{4.20}$$

where n_0' and n_1' are the number of occurrence of 0 and 1 in the training set, n_0 and n_1 are the counts from the input. The first term $l_m(\alpha n_0' + n_0, \alpha n_1' + n_1)$ is the adaptive code length based on the counts from the training set plus the input. The second term $l_m(\alpha n_0', \alpha n_1')$ is estimated based on the training set alone. α is a parameter indicating how much we can trust the training set statistics. Our goal is to minimize the difference between these two terms which represents the mismatch between the training set and the input.

Once the context quantizer is predesigned, actual coding remain the same as described in Section 4.5.3

4.6.2. Implementation

The adaptive code lengths used in the dynamic programming are calculated based on the counts, (n_0, n_1), for each possible context cell (quantizer interval). Since the dynamic programming algorithm uses the actual adaptive code length for a given finite sequence and a fixed δ as the cost function, it can automatically decide the optimum number of coding contexts *M* This is simply done by increasing the number of context quantizer cells in the bottom-up dynamic programming process, until reaching the point where the actual code length starts to increase.

Given a quantizer interval and the associated 0 and 1 counts, the corresponding adaptive code lengths can be computed in $O(1)$ time independent of the interval length by a fast algorithm proposed in [56]. The idea is to use look-up table to compute the adaptive code length for small values of count, and use Stirlings approximation for large values when such an approximation yields high precision. With the fast adaptive codelength computation technique, one can precompute and store the adaptive code lengths for all possible quantizer intervals. This preprocess takes $O(N^2)$ time, where N is the number of distinct unquantized raw contexts Aided by the intermediated results of the preprocess (adaptive code lengths of all possible quantizer intervals), the dynamic programming algorithm can be completed in $O(MN^2)$ time.

Another technique to speed up the dynamic programming algorithm is to merge all the raw contexts that have the same counts. This can significantly reduce the number of initial contexts subject to quantization. This will not affect the optimal solution because those contexts would be merged anyways by the CQ scheme above.

The estimator (4.13) is optimal if the events in a context are independent and the prior distribution initially is beta distributed with nuisance parameter δ. In this view all the contexts of the same counts have the same distribution of the parameter $\hat{P}_{x|c}(1 \mid \mathbf{c})$, which also suggests that they should be quantized into the same context cell.

4.6.3. Experimental Results

We implemented the proposed minimum adaptive codelength context quantization scheme as described above. In order to evaluate the performance of the scheme, we compare it with MDL-based method, JBIG and JBIG2 standard on a set of twelve halfttone images. The results are shown in Table 4.4. As you can see, the minimum adaptive codelength context quantization is superior to the other schemes including MDL-based context quantization method, image dependent scheme and two standards, JBIG and JBIG2. The average compression gains over JBIG by 17% and JBIG2 by 8%; while the peak compression improves 24% to JBIG and 11% to JBIG2. However, there is not much improvement when comparing image dependent method and minimum adaptive codelength method. The explanation of this is image dependent method can achieve the code length very close to the optimal one by adaptively updating the context quantizer output on the fly.

Table 4.4 Bit rate comparison between minimum mismatch CQ by adaptive code length scheme and other schemes

Image	*Minimum Adaptive codelength*	*Image Dependent CQ*	*MDLCQ*	*JBIG2*	*JBIG*
Barbara	0.713	0.713	0.722	0.770	0.867
Lena	0.675	0.674	0.686	0.744	0.884
Mandrill	0799	0.800	0.806	0.841	0.927
Boat	0.703	0.704	0.714	0.738	0.772
Clown	0.495	0.496	0.503	0.535	0.605
Goldhill	0.665	0.665	0.669	0.725	0.827
Grandma	0.660	0.660	0.666	0.732	0.820
Lynda	0.564	0.564	0.565	0.631	0.73
Couple	0.706	0.706	0.708	0.759	0.853
Tiffany	0.571	0.572	0.574	0.612	0.623
Cameraman	0.640	0.640	0.654	0.682	0.703
Man	0.553	0.553	0.560	0.601	0.686
Sum	7.744	7.747	7.827	8.370	9.297

4.7. Conclusions

Context quantizer is an effective technique to alleviate context dilution problem in conditional entropy coding. Up to now, all the context quantizers are optimized with respect to the statistics of a training set. An ensuing question is how to handle any mismatch in statistics between the training set and the input image. Unless they match perfectly, there exists an optimal blend of the statistics to achieve the minimum adaptive code length. Three algorithms are proposed in this chapter to handle this problem.

MDL-based algorithm is to minimize the sum of the bits emitted by the conditional entropy coder using the context quantizer and the side information to describe the context space partition. This side information is the compressed state sequence of rare context instances by entropy coding. Image dependent context quantizer is a MCECQ designed based on input statistics alone. The cost of the side information is low since only context quantizer centroids are transmitted. An efficient method to handling the rare context instances is proposed. The conditional probability are initialized from the training set and adaptively updated on the fly. As a result, more accurate context quantizer outputs will be generated to drive the arithmetic coder and finally the compression efficiency will be improved. Minimum adaptive code length context quantizer is aiming to minimize the effect of mismatch of the statistics between the training set and the input. The actual adaptive code length difference between the two sets, the training set plus the input and the training set alone, is minimized.

The main difference between MDL-based algorithm and the other two methods is the way of dealing with the rare contexts. In MDL-based method, the context quantizer outputs for rare context instances are transmitted as side information. In image dependent context quantizer and minimum adaptive code length context quantizer design schemes, the conditional probabilities of rare context instances are first estimated using the training set under lower order context template definition and then adaptively being updated on the fly. These conditional probabilites are compared with the centroids of the context quantizer to obtain the output index. When the estimates of conditional probability of these rare context instances are inaccurate and they also do not occur very often in the input image, MDL-based method will show its advantage over the other two methods. Because in this case, the conditional probabilities of these rare context instances turn to be peaky, which will lead to small data rate. Sending side information using ECSS method will become a better choice.

Not surprisingly, all these three approaches outperform both JBIG and JBIG2 standard. The minimum adaptive code length context quantization scheme and the image dependent with efficient side information scheme achieve the best performance on the chosen data set with peak compression improvement of 24% over JBIG and 11% over JBIG2. When the statistics mismatch of the training set and input image is moderate, image dependent method can achieve the code length very close to the optimal one by adaptively updating the context quantizer output on the fly. The context quantizer design algorithm in the image dependent method is MCECQ based only on the input image statistics. Its complexity and computational cost is lower than minimum adaptive code length context quantizer design scheme, which is aiming to minimize the effect of mismatch of the statistics between the training set and the input in any circumstance.

CHAPTER 5.
CONTEXT BASED CLASSIFICATION AND QUANTIZATION

5.1 Motivation

Various classification techniques have been shown to be effective in adaptive quantization schemes in wavelet image coders. These classification-based schemes [53][57-61] separate the subband data into several subsequences with different distributions. A set of quantizers are customized to the individual distribution components. Then a sequence of quantizers can be used rather than a single average quantizer fitted to the overall input statistics. Thus the quantization scheme can be adapted to the classification information and a better performance can be expected. The difference among various algorithms mainly lies in the approach of modeling the mixture of subsources with different statistical characteristics. The methods proposed in [60] are all based on block-wise classification. Each class is characterized as a generalized Gaussian source with different parameter. A different quantizer is then used for each class. Context-based classification method is to assign a class to each subband coefficient based on the causal and quantized spatial neighbourhood context. Two schemes based on this principle are presented in [61] and [53]. In both methods, a parametric distribution model for each class is assumed. Generalized Gaussian distribution (GGD) is used in [61], while simpler Laplacian model is used in [53]. According to the estimated parameters of each class, the best quantizer chosen from a set of available quantizers is assigned. The advantage of the parametric modeling approach is the small modeling cost. However, it can result in significant loss in the coding efficiency in the case of mismatched statistics.

5.2 Non-Parametric Context Based Classification and Quantization

5.2.1 Basic Idea

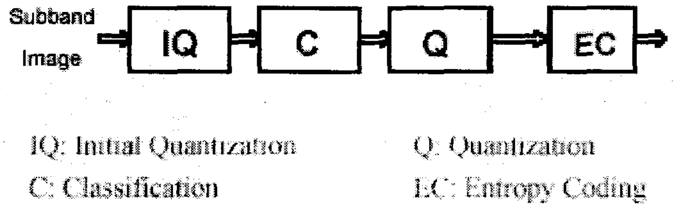


Figure 5-1 Diagram for Proposed Scheme

The whole procedure of our context-based classification and adaptive quantization scheme is shown in Figure 5-1. First, we preprocess all data using a single initial quantizer with its rate close to the target bit rate. Then the context is defined by the previous quantized coefficients. The context model is used to accurately estimate the probability mass function (pmf) of the upcoming symbol. This pmf estimate is generally made by counting the number of occurrences of symbol in each context and computing the relative frequencies. The counts can be organized into a histogram for each context, with one entry per symbol. Each of these pmfs can be viewed as an estimation of the true probability density distribution (pdf) of the coefficient in the corresponding context. The coefficients with similar pmf estimates will be assigned to the same class. We applied the histogram quantization scheme proposed in section 3.1.2. to make an effective classification based on the pmf estimates. According to the statistics of each class, the best quantizer will be chosen from a set of uniform dead zone threshold quantizers. Bit allocation among the classes is also performed at the same time to optimize the overall rate distortion performance. The classic bit allocation method based on [62] is applied in our scheme. An iterative bit allocation algorithm is used to determine the Lagrange multiplier λ in the cost function $J = D + \lambda R$, where D and R are the overall distortion and rate. For each λ the algorithm selects the quantizer which minimizes J. The iteration on λ terminates until the algorithm finds $J = J^*$ for which the R falls within a certain range of the target rate. In the above scheme, the side information includes the description of the classification map and the quantizer information for each class. We adopt entropy coded state sequence method proposed in Section 3.2.2.2 to code the classification map.

5.2.2 Selection of Initial Quantizer

The initial quantizer is selected from the set of available uniform threshold dead zone quantizers. The choice of the initial quantizer affects the accuracy of the classification and the amount of the side information spent on describing the classification map. A good initial quantizer should meet the following three criteria. First, its rate is close to the target bit rate for the classification varies with the bit rate. The classification map obtained for a low bit rate initial quantizer will not be suitable for high bit rate coding. Secondly, a better rate distortion performance of the initial quantizer leads to a more accurate classification. Finally, another preferred feature of the initial quantizer is the small size of the symbol set. This feature can save bits of the side information spent on describing the classification map.

5.2.3 Context Based Classification

In our proposed scheme, the classification is made based on the estimated pmf of the coefficients. As a result, the coefficients associated with the same context instance, which is defined by the initial quantizer, will be assigned to the same class because they share the same pmf. Although the simplest way is to define each context instance as a single class, the number of the different contexts instances could be very large especially if a high resolution initial quantizer is used. Because the quantizer information for each class needs to be sent as side information, it is obviously inefficient to have too many classes. Therefore the number of the classes should be reduced. A natural solution is to merge the context instances with similar pmf's until a desired number of the classes, N , is achieved. With this motivation, we apply the histogram quantization method described in section 3.1.2, in which the dissimilarity of the pmfs is measured by the relative entropy.

Clearly, the classification map needs to be compressed to minimize the size of side information. Entropy coded state sequence method, which is presented in Section 3.2.2.2, is applied here to code the classification map. Instead of sending class index for each of the coefficient, the classification indices of the different context instances are sent in the same order that happens in the input sequence. Both the encoder and the decoder will build a context book on the fly to memorize the classification information of the contexts that have already appeared. When the context instances reoccur, it is not necessary to transmit their classification indices. At the end of coding the input sequence, this context book, built at both encoder and decoder, will be the same as the original one which is not actually transmitted. Moreover, the classification map can be further compressed using entropy coding by exploiting the fact that the population of different groups is unequal.

One issue to be addressed is that in order to construct the above classification map, our choice of the quantizer set, among which the best fit quantizer will be chosen for each class, will be limited. The reduced classification map is built based on the context under the definition of initial quantizer. Therefore in order to obtain the accurate classification information, the context should be correctly calculated. However, in the real coding process, the past context data is already requantized by the optimal quantizer chosen from the quantizer set. As a result, we need to put a restriction on the quantizer set so that the reconstruction values using the real quantizer will fall into the same cell as the original data under the definition of the initial quantizer. This condition will guarantee that the context information under the initial quantizer can be recovered at the decoder.

5.3 Experimental Results

To demonstrate the effectiveness of the above techniques we perform some experiments on the 512×512 Barbara image processed by a wavelet transform to a depth of three, where the filter set is the standard 9-7 configuration [54].

First we generate the operational rate-distortion curve (RD curve) for the subband image data using the available uniform threshold dead zone quantizer set. The candidates for the initial quantizers will have two features. The one is that its (R , D) values are close to the above rate distortion curve. The other one is that it should be with small number of cells. The context template was defined as the four causal nearest neighbors. After we quantize the data using the selected initial quantizer, the histogram quantization method is applied to perform the classification. Then the bit allocation is performed among the classes. For each of the Lagrange multiplier λ, we select the quantizer for each class to minimize the cost function $J = D + \lambda R$. By changing the λ, a new rate distortion curve is produced.

Figure 5.2 and Figure 5.3 show the results for subband LH_2, a 128×128 highpass image. Figure 5.4 is for subband LH_2 In Figure 5.2, two RD curves, obtained with two different initial quanizers are compared with the original RD curve which is produced without classification. The two initial quantizers are chosen around 0.2bpp and 0.5bpp seperately. The performance of classification scheme is always better than original one. The RD curve with initial bit rate at 0.5bpp can not achieve the low bit rate that is because of the scheme we used for coding classification map. At high bit rate the RD curve with initial bit rate at 0.5bpp always outperforms the 0.2bpp one. The complete RD curve can be obtained by combining the pieces with the best performance associated with different initial quantizers at all rate. Figure 5.3 and Figure 5.4 showed the complete RD curve for subband at 0e ecause the scheme we used for codingThe one the piecwithinitial it rate of 0.5bpp can not achieve the low bit rate that is because of the requirement of the scheme we used for coding the classification map. At high bit rate the adaptive quantizer with initial bit rate of 0.5bpp always outperform than the 0.2bpp one. The complete RD curve can be obtained b y combining the pieces with the best performance associated with different initial quantizers at all bit rates. Figure 5.3 and Figure 5.4 show the resulting RD curve for subband LH_2 and subband HL_3.

The experiments also showed that the side information of the system could be kept at a very low level. For example, when coding the subband HL_3 at bit rate of 0.4bpp, the side information rate is as low as 0.0037bpp.

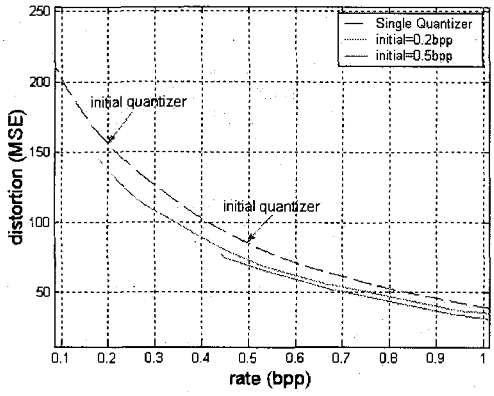


Figure 5-2 Rate Distortion Curve for Different Initial Quantizer for LH_2 Subband

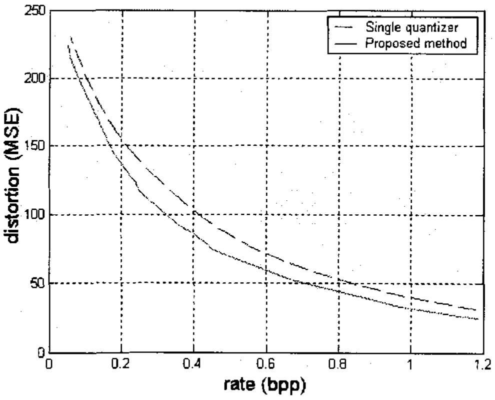


Figure 5-3 Overall rate-distortion curve for LH_2 Subband

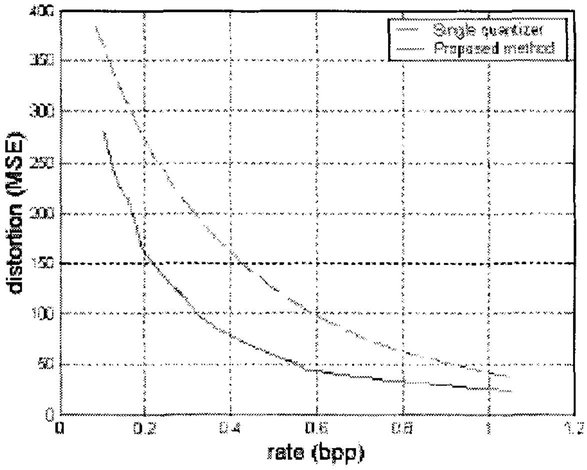


Figure 5-4 Overall rate-distortion curve for *LH_3* Subband

5.4 Conclusions

This section has presented a new scheme for context based classification and adaptive quantization. Our method employs histogram quantization technique to perform the classification. Since a non-parametric model is used, the statistics of each class can be estimated more accurately, and therefore, a better quanzation scheme can be applied to improve the overall rate distortion performance. It is also shown that the side information is kept at a relative low level.

CHAPTER 6.
CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this dissertation, we develop new techniques for context quantization under the criteria of minimum conditional entropy, minimum adaptive code length, and for optimal rate-distortion performance in image compression. The following contributions have been made.

First, we propose a context quantization scheme to alleviate the context dilution problem in high-order context-based adaptive entropy coding. Our method employs a histogram quantizer to reduce a large set of all possible context instances to a manageable number of coding states. The resulting context quantizer is proved to be (locally) optimal in the sense of minimizing the conditional entropy.

The real challenge to apply the above optimal context quantizer in practice is how to describe the resulting complex partition of the context space. We then tackle this problem. Two novel methods are proposed. Coarse context quantization method is to decrease the size of the context book by preprocessing the context space. Entropy coded state sequence method is to reduce the bits for coding the individual entry of the context book. The experiments show their effectiveness in compressing the side information.

The context quantizer design for binary source is studied next. The probability simplex space of binary source is one dimensional. This reduces context quantizer design to a scalar quantizer design problem and the global optima can be achieved by dynamic programming. Currently all the context quantizers described in the literature are designed off-line and being optimized with respect to the statistics of the training set. We propose three novel schemes to deal with the mismatches between the training set and the input image. MDL-based context quantizer is to minimize the sum of the bits of coding the input and the side information to describe the context quantizer mappings. Image dependent context quantizer is a MCECQ with an efficient way of handling the rare context instances. Finally, minimum adaptive code length context quantizer is to minimize the effect of mismatch between the input and the training set statistics. The difference between the two sets, the training set plus the input and the training set alone, is minimized. Our schemes superior to JBIG and JBIG2 on the chosen set of twelve halftone images with the peak compression improvement of 24% and 11% and average gains of 17% and 8%.

Finally, we extend our work to the joint design of both quantizers and entropy coders. A non-parametric modelling context-based classification and adaptive quantization scheme on coefficient basis is presented. A finite state quantizer and entropy coder are produced with the same procedure. The results show that it has great potential to improve the overall compression system performance.

6.2 Future Work

6.2.1 Context Shape Optimization

The context quantizer design we discussed in this thesis require a fixed context space. The definition of the initial context space has significant impact on the overall compression efficiency. If the initial context space is defined to be too large, the computation complexity will be highly increased without any benefit to the compression performance. However, if the context space is defined to small, some important information is already missed even before applying any context quantization scheme. Context shape optimization is a challenging and interesting problem, which has remained largely untreated.

6.2.2 Application to Other Image/Video Codec

Although the context based adaptive entropy coding and quantization techniques presented in this thesis are studied in the context of image compression, they can also be used to improve the video compression system performance. When demonstrating the power of our schemes, the experiments are performed on wavelet subband images and binary halftone images. However, these techniques can be applied to any other type of images. When exploiting the context quantizer techniques designed for binary source on non binary source, we need to decompose it into a sequence of binary decisions and coded using the proposed method.

6.2.3 Application to Distributed Multimedia Compression

The problem of data compression with side information at the decoder appears in numerous practical applications such as distributed sensor systems, network communications, stereo and multi-camera systems and surveillance systems [63]. To compress the side information, one needs to build quantizers that minimize the rate at which the source can he encoded with a constraint on the entropy of the quantized side information. This problem is strongly related to our context quantizer design problem. Therefore our context quantization techniques presented in this thesis can be applied to a wide range of distributed multimedia compression problems.

BIBLIOGRAPHY

[1] K. Sayood, *Introduction to Data Compression.* ,third ed.2006,

[2] J. Vaisey and Tong Jin, "An iterative algorithm for context selection in adaptive entropy coders," in *Proceedings of ICIP 2002 International Conference on Image Processing, 22-25 Sept. 2002,* 2002, pp. 93-6.

[3] T. Jin and J. Vaisey, "Efficient side-information context description for context-based adaptive entropy coders," in *Proceedings. DCC 2004. Data Compression Conference, 23-25 March 2004,* 2004, pp. 543.

[4] T. Jin, X. Wu and J. Liang, "Context Quantizer Design for Minimum Adaptive Arithmetic Code Length Using Preknowledge," *IEEE Trans. Image Process.,* Submitted. Oct. 2006.

[5] T. Jin and X. Wu, "MDL Based Adaptive Context Quantization," *Picture Coding Symposium,* 2004.

[6] T. Jin and J. Vaisey, "A New Method for Context Based Classification and Adaptive Quantization in Subband Image Coding," *Picture Coding Symposium,* 2004.

[7] C. E. Shannon, "A mathematical theory of communication," *Bell Syst Tech. J,* vol. 27, pp. 379–423, 1948.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley New York, 1991,

[9] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc.IRE,* vol. 40, pp. 1098-1101, 1952.

[10] W. B. Pennebaker, *Jpeg: Still Image Data Compression Standard.* Kluwer Academic Publishers, 1993,

[11] M. Weinberger, G. Seroussi and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles andstandardization into JPEG-LS," *Image Processing, IEEE Transactions on,* vol. 9, pp. 1309-1324, 2000.

[12] J. Shapiro, D. S. R. Center and N. Princeton, "Embedded image coding using zerotrees of wavelet coefficients," *Signal Processing, IEEE Transactions*

on [See also Acoustics, Speech, and Signal Processing, IEEE Transactions on], vol. 41, pp. 3445-3462, 1993.

[13] J. Rissanen and G. G. Langdon Jr, "Universal modeling and coding," *IEEE Trans. Inf. Theory,* vol. IT-27, pp. 12-23, 01/. 1981.

[14] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning inhierarchical trees," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 6, pp. 243-250, 1996.

[15] G. J. Sullivan and T. Wiegnad, "Video Compression—From Concepts to the H. 264/AVC Standard," *Proc IEEE,* vol. 93, pp. 18-31, 2005.

[16] D. Marpe, T. Wiegand and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Communications Magazine,* vol. 44, pp. 134-43, 2006.

[17] G. Bjontegaard and K. Lillevold, "Context-Adaptive VLC Coding of Coefficients," *JVT Document JVT-C028, Fairfax, VA, may,* 2002.

[18] S. W. Golomb, "Run-length encodings," *IEEE Trans. Inf. Theory,* vol. IT-12, pp. 399-401, 1966.

[19] ITU-T Recommendation T.88, "Information technology - Lossy/Lossless coding of Bi-level Images," March 2000.

[20] P. Howard, F. Kossentini, B. Martins, S. Forchhammer and W. Rucklidge, "The emerging JBIG2 standard," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 8, pp. 838-848, 1998.

[21] G. K. Wallace, "The JPEG still picture compression standard," *Commun ACM,* vol. 34, pp. 30-44, 1991.

[22] Anonymous "JBIG," *Http://www.Jpeg.org/jbig/index.Html,*

[23] J. Standard, "Coded Representation of Picture and Audio Information-Progressive Bi-level Image Compression Standard," *ISO/IEC JTC1/SC29/WG9,* 1990.

[24] I. Hontsch and L. J. Karam, "Locally adaptive perceptual image coding," *IEEE Trans. Image Process.,* vol. 9, pp. 1472-83, 2000.

[25] Xiaolin Wu and N. Memon, "Context-based lossless interband compression-extending CALIC," *IEEE Trans. Image Process.,* vol. 9, pp. 994-1001,2000.

[26] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *Communications, IEEE Transactions on,* vol. 45, pp. 437-444, 1997.

[27] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Process.,* vol. 3, pp. 572-88, 1994.

[28] A. Moffat, R. M. Neal and I. H. Witten, "Arithmetic coding revisited," *ACM Transactions on Information Systems (TOIS),* vol. 16, pp. 256-294, 1998.

[29] P. G. Howard and J. S. Vitter, *Practical Implementations of Arithmetic Coding.* Brown University, Dept. of Computer Science, 1991,

[30] W. Pennebaker, J. Mitchell, G. Langdon Jr and R. Arps, "An overview of the basic principles of the Q-Coder adaptive binary arithmetic coder," *IBM Journal of Research and Development,* vol. 32, pp. 717-726, 1988.

[31] M. J. Weinberger, J. J. Rissanen and R. B. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. Image Process.,* vol. 5, pp. 575-86, 1996.

[32] J. Rissanen, "A universal data compression system," *IEEE Trans. Inf. Theory,* vol. IT-29, pp. 656-64, 1983.

[33] F. M. J. Willems, Y. M. Shtarkov and T. J. Tjalkens, "The context-tree weighting method: basic properties," *IEEE Trans. Inf. Theory,* vol. 41, pp. 653-64, 1995.

[34] N. Ekstrand, "Lossless compression of grayscale images via context tree weighting," in *Proceedings of Data Compression Conference - DCC '96, 31 March-3 April 1996.*

[35] M. Arimura, H. Yamamoto and S. Arimoto, "A bitplane tree weighting method for lossless compression of gray scale images," *IEICE Trans. Fund. Electron. Commun. Comput. Sci.,* vol. E80-A, pp. 2268-71, 1997.

[36] M. Mrak, D. Marpe and T. Wiegand, "A context modeling algorithm and its application in video compression," in *Proceedings of International Conference on Image Processing, 14-17* Sept. 2003, pp. 845-8.

[37] Anonymous "Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression," *CCITT Draft Recommendation T. 82, ISO/IEC Draft International Standard 11544,* Apr. 1992.

[38] X. Wu, "Context quantization with fisher discriminant for adaptive embedded wavelet image coding," in *Proceedings of Conference on Data Compression (DCC'99), 29-31* March 1999

[39] Xiaolin Wu, P. A. Chou and Xiaohui Xue, "Minimum conditional entropy context quantization," in *2000 IEEE International Symposium on Information Theory, 25-30 June 2000,* pp. 43.

[40] S. Forchhammer, Xiaolin Wu and J. D. Andersen, "Optimal context quantization in lossless compression of image data sequences," *IEEE Trans. Image Process.,* vol. 13, pp. 509-17, 2004.

[41] Jianhua Chen, "Context modeling based on context quantization with application in wavelet image coding," *IEEE Trans. Image Process.,* vol. 13, pp. 26-32, 2004.

[42] Zhen Liu and L. J. Karam, "Mutual information-based analysis of JPEG2000 contexts," *IEEE Trans. Image Process.,* vol. 14, pp. 411-22, 2005.

[43] Mantao Xu, Xiaolin Wu and P. Franti, "Context quantization by kernel Fisher discriminant," *IEEE Trans. Image Process.,* vol. 15, pp. 169-77, 2006.

[44] D. Taubman, "High performance scalable image compression with EBCOT," *Image Processing, IEEE Transactions on,* vol. 9, pp. 1158-1170, 2000.

[45] D. Greene, F. Yao and Tong Zhang, "A linear algorithm for optimal context clustering with application to bi-level image coding," in *Proceedings of IPCIP'98 International Conference on Image Processing, 4-7 Oct. 1998,* pp. 508-11.

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

[47] Xiaolin Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Trans. Image Process.,* vol. 6, pp. 656-64, 1997.

[48] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," *Signals, Systems & Computers, 1997.Conference Record of the Thirty-First Asilomar Conference on,* vol. 2, 1997.

[49] Xiaolin Wu, Jiang Wen and Wing Hung Wong, "Conditional entropy coding of VQ indexes for image compression," *IEEE Trans. Image Process.,* vol. 8, pp. 1005-13, 1999.

[50] Zixiang Xiong, Xiaolin Wu, S. Cheng and Jianping Hua, "Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms," *IEEE Trans. Med. Imaging,* vol. 22, pp. 459-70, 2003.

[51] Zixiang Xiong, Xiaolin Wu, D. Y. Yun and W. A. Pearlman, "Progressive coding of medical volumetric data using three-dimensional integer wavelet packet transform," in *Visual Communications and Image Processing '99, 25-27 Jan. 1999,* pp. 327-35.

[52] C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," in *Proceedings DCC '97. Data Compression Conference, 25-27 March 1997,* pp. 241-50.

[53] Youngjun Yoo, A. Ortega and Bin Yu, "Image subband coding using context-based classification and adaptive quantization," *IEEE Trans. Image Process.,* vol. 8, pp. 1702-15, 1999.

[54] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.,* vol. 1, pp. 205-20, 1992.

[55] R. E. Bellman, *Dynamic Programming.* Dover Publications, 2003,

[56] B. Martins and S. Forchhammer, "Tree coding of bilevel images," *IEEE Trans. Image Process.,* vol. 7, pp. 517-28, 1998.

[57] R. L. Joshi, T. R. Fischer and R. H. Bamberger, "Optimum classification in subband coding of images," in *Proceedings of 1st International Conference on Image Processing, 13-16 Nov. 1994,* pp. 883-7.

[58] Y. Yoo, A. Ortega and Bing Yu, "Adaptive quantization of image subbands with efficient overhead rate selection," in *Proceedings of 3rd IEEE International Conference on Image Processing, 16-19 Sept. 1996,* pp. 361-4.

[59] B. Yu, "A statistical analysis of adaptive quantization based on causal past," in *Proceedings of 1995 IEEE International Symposium on Information Theory, 17-22 Sept. 1995,* pp. 375.

[60] R. L. Joshi, H. Jafarkhani, J. H. Kasner, T. R. Fischer, N. Farvardin, M. W. Marcellin and R. H. Bamberger, "Comparison of different methods of classification in subband coding of images," *IEEE Trans. Image Process.,* vol. 6, pp. 1473-86, 1997.

[61] S. M. LoPresto, K. Ramchandran and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proceedings DCC '97. Data Compression Conference, 25-27 March 1997,* pp. 221-30.

[62] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers [speech coding]," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 36, pp. 1445-53, 1988.

[63] J. Cardinal, "Compression of side information," in *2003 IEEE International Conference on Multimedia and Expo, 6-9 July 2003,* pp. 569-72.

[64] Nemhauser, "*Introduction to Dynamic Programming*" 1966.