DCT-based Image/Video Compression: New Design Perspectives

by

Chang Sun

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

To push the envelope of DCT-based lossy image/video compression, this thesis is motivated to revisit design of some fundamental blocks in image/video coding, ranging from source modelling, quantization table, quantizers, to entropy coding. Firstly, to better handle the heavy tail phenomenon commonly seen in DCT coefficients, a new model dubbed transparent composite model (TCM) is developed and justified. Given a sequence of DCT coefficients, the TCM first separates the tail from the main body of the sequence, and then uses a uniform distribution to model DCT coefficients in the heavy tail, while using a parametric distribution to model DCT coefficients in the main body. The separation boundary and other distribution parameters are estimated online via maximum likelihood (ML) estimation. Efficient online algorithms are proposed for parameter estimation and their convergence is also proved. When the parametric distribution is truncated Laplacian, the resulting TCM dubbed Laplacian TCM (LPTCM) not only achieves superior modeling accuracy with low estimation complexity, but also has a good capability of nonlinear data reduction by identifying and separating a DCT coefficient in the heavy tail (referred to as an outlier) from a DCT coefficient in the main body (referred to as an inlier). This in turn opens up opportunities for it to be used in DCT-based image compression.

Secondly, quantization table design is revisited for image/video coding where soft decision quantization (SDQ) is considered. Unlike conventional approaches where quantization table design is bundled with a specific encoding method, we assume optimal SDQ encoding and design a quantization table for the purpose of reconstruction. Under this assumption, we model transform coefficients across different frequencies as independently distributed random sources and apply the Shannon lower bound to approximate the rate distortion function of each source. We then show that a quantization table can be optimized in a way that the resulting distortion complies with certain behaviour, yielding the so-called optimal

distortion profile scheme (OptD). Guided by this new theoretical result, we present an efficient statistical-model-based algorithm using the Laplacian model to design quantization tables for DCT-based image compression. When applied to standard JPEG encoding, it provides more than 1.5 dB performance gain (in PSNR), with almost no extra burden on complexity. Compared with the state-of-the-art JPEG quantization table optimizer, the proposed algorithm offers an average 0.5 dB gain with computational complexity reduced by a factor of more than 2000 when SDQ is off, and a 0.1 dB performance gain or more with 85% of the complexity reduced when SDQ is on.

Thirdly, based on the LPTCM and OptD, we further propose an efficient non-predictive DCT-based image compression system, where the quantizers and entropy coding are completely re-designed, and the relative SDQ algorithm is also developed. The proposed system achieves overall coding results that are among the best and similar to those of H.264 or HEVC intra (predictive) coding, in terms of rate vs visual quality. On the other hand, in terms of rate vs objective quality, it significantly outperforms baseline JPEG by more than 4.3 dB on average, with a moderate increase on complexity, and ECEB, the state-of-the-art non-predictive image coding, by 0.75 dB when SDQ is off, with the same level of computational complexity, and by 1 dB when SDQ is on, at the cost of extra complexity. In comparison with H.264 intra coding, our system provides an overall 0.4 dB gain or so, with dramatically reduced computational complexity. It offers comparable or even better coding performance than HEVC intra coding in the high-rate region or for complicated images, but with only less than 5% of the encoding complexity of the latter. In addition, our proposed DCT-based image compression system also offers a multiresolution capability, which, together with its comparatively high coding efficiency and low complexity, makes it a good alternative for real-time image processing applications.

Acknowledgements

Before a long list of people I am indebted for making this thesis possible, my deepest appreciation and gratitude is reserved for my supervisor, Professor En-hui Yang, for his invaluable and constant guidance through out my Ph.D. studies at the University of Waterloo. Through years of extensive training from Professor Yang, I not only became inspired to do cutting-edge research, but also profoundly influenced to think more precisely, logically, and sharply. Beyond the academic training, he also provides care to my personal life and empowers me to persevere through all expected, inevitable, and unforeseeable obstacles.

I am extremely grateful to all my examining committee members, formed by distinguished scholars. I would like to thank Professor Zhou Wang, Professor George Freeman, and Professor Xinzhi Liu, for their valuable comments for my comprehensive exam and their commitment to my Ph.D. thesis defence. And I would also like to thank Professor Jie Liang from the Simon Fraser University, for his commitment to serve as my external examining committee member.

My sincere thanks also goes to previous and current members of the Multimedia Communications Laboratory, many of whom I have forged strong friendships and collaborations with, including Dr. Da-ke He, Dr. Wei Sun, Dr. Xiang Yu, Dr. Jin Meng, Dr. Lin Zheng, Dr. Mehdi Torbatian, Dr. Shenghao Yang, Krzysztof Hebel, James Ho, Yuhan Zhou, Hui Zha, Fei Teng, Jie Zhang, Krishna Rapaka, Nan Hu, Yueming Gao, and Mahshad Eslamifar.

Finally, I give my gratitude to my father, Guangyu Sun, my mother, Ruimin Ge, and my wife, Huijuan Ding, for their unconditional love, understanding, encouragement and support to my Ph.D. studies.

To Huijuan and Ethan

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Thesis motivation

Digital image/video has been a ubiquitous and essential component of the entertainment, broadcasting, and communications industries, and image/video data increases explosively over the years. According to a Cisco report [11], global mobile data traffic reached 885 petabytes per month in 2012, among which 50% was video traffic; by 2017, the global mobile data traffic would increase 13-fold and the percentage of video traffic would further increase to two-third. For images, over 1 trillion photos are estimated online at any given time, from web pages, social media, ad photos, etc. This fuels a demand for image/video compression to pursue the best possible coding efficiency for accommodating the rapidly growing demand for image/video data.

For catering this demand, many image/video coding standards have been developed during the past decades [1], [5], [30], [31], [61], [32], [51]. Most of these standards adopt transform coding to convert pixel values to transform coefficients in order to de-correlate

the input image/video signal. The transform coding exploits the spacial redundancy in an image/video frame, leaving the same information more concentrated in certain frequencies in the transform domain than it was in the spacial domain. Among many transforms, the discrete cosine transform (DCT) is a very popular one due to its capability to yield a good tradeoff between complexity and energy compaction performance, while easily fitting into the block-matching motion estimation framework [31], [32] for video coding. To name a few, JPEG [1], [58], H.264/AVC [31], [61], and the newly developed HEVC [32], [51] are all DCT-based image/video coding methods.

Given an $N \times M$ image block, I, in the spatial domain, the pixel at coordinates (x, y) is denoted $I_{x,y}$. The two-dimensional (2-D) $N \times M$ DCT to transform I into an image in the frequency domain or DCT domain, C, is defined by

$$C_{u,v} = \frac{2}{\sqrt{NM}} F(u)F(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I_{x,y} cos \frac{(2x+1)u\pi}{2N} cos \frac{(2y+1)v\pi}{2M}, \tag{1.1}$$

for $u = 0, 1, \cdots, N-1$, $v = 0, 1, \cdots, M-1$, where $F(t) = 1/\sqrt{2}$ for $t = 0$, or $F(t) = 1$ otherwise. The corresponding 2-D inverse DCT is given by

$$I'_{x,y} = \frac{2}{\sqrt{NM}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u)F(v)C_{u,v} cos \frac{(2x+1)u\pi}{2N} cos \frac{(2y+1)v\pi}{2M}, \tag{1.2}$$

for $x = 0, 1, \cdots, N-1$, $y = 0, 1, \cdots, M-1$.

Prediction is a method to remove the spatial and/or temporal redundancy in the input signal. A DCT-based image/video compression system can either be non-predictive such as in JPEG, or it can be predictive such as in H.264 and HEVC. A DCT-based non-predictive image compression system is shown in Fig. 1.1. The input image is first partitioned into blocks, and DCT is applied to each block. Then DCT coefficients are quantized (typically by uniform scaler quantizers) and finally entropy coded. A DCT-based predictive video

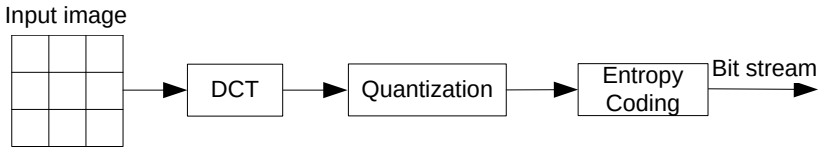

Figure 1.1: Block diagram of a typical DCT-based non-predictive image compression system.

compression system is illustrated in Fig. 1.2, where intra prediction is used to remove the spatial redundancy in an individual video frame and inter prediction is used to remove the temporal redundancy among adjacent video frames. The predicted signal is subtracted from the input signal, and the residue is then transformed by DCT. DCT coefficients are quantized and finally entropy coded. In addition, in-loop filters are usually applied to the reconstruction signal to mitigate the blocking artifacts of the decoded video frames, and the resulting reconstructed frames are stored in the decoded picture buffer as reference video frames. Prediction generally improves coding efficiency, however, it significantly increases encoding complexity and makes the encoder (such as the HEVC encoder as shown in [9] and Chapter 4 of this thesis) generally hard to be applied to real-time applications. As such, non-predictive coding methods such as JPEG are still widely used as of this writing due to their lower complexity. The scope of this thesis covers the DCT-based non-predictive coding, with the predictive case left for future work (note that the methodology proposed in this thesis can be extended to predictive coding as discussed in the future work in Chapter 5).

To further push the envelope of DCT-based (lossy) image/video compression, this thesis is motivated to revisit some of the most important design blocks/components–from source modelling, to quantization table design, to quantizers and entropy coding design, and further proposes some new design perspectives, which may provide the essential theoretical guidance and/or practical methodology to the next generation image/video coding

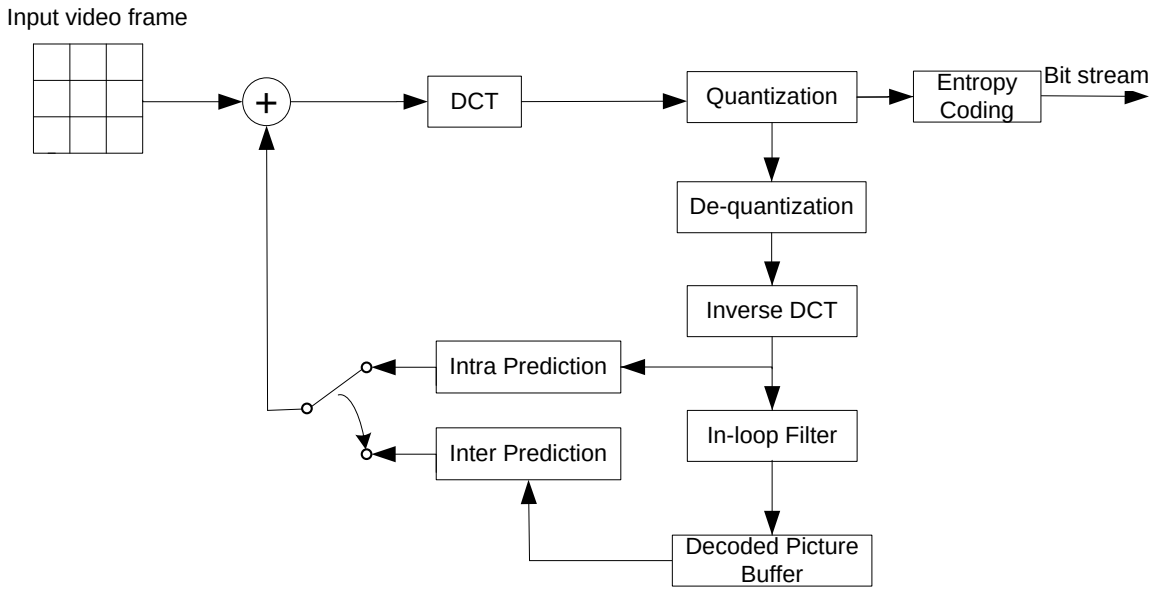

Figure 1.2: Block diagram of a typical DCT-based video compression system.

standard.

To be specific, work in this thesis is mainly motivated by a desire to answer the following three questions.

1) How can we effectively and efficiently model DCT coefficients for DCT-based image/video compression?

The popularity of DCT has attracted, during the past a few decades, a lot of interest in understanding the statistical distribution of DCT coefficients [43], [33], [36], [25], [6], [50], [21], [76], [75]. Deep and accurate understanding of the distribution of DCT coefficients would be helpful to the design of lossy image/video compression algorithms [50], [21]. Many probability prototypes were proposed in the literature to model DCT coefficients, including Gaussian [6], generalized Gaussian [33], [25], Laplacian [43], [36], [25], [50], and Cauchy [21].

However, those models either suffer from poor modelling accuracy that could not fit the histogram very well, or high complexity for parameter estimation that could not be applied online in practice. As such, further studies and revisit on the distribution of DCT coefficients are suggested, as we raise the question–how to effectively and efficiently model DCT coefficients, such that they can be accurately described by the statistical model that can be obtained online with low complexity for parameter estimation?

2) With a statistical model, how can we efficiently design a quantization table in the rate-distortion (RD) sense for DCT-based image/video compression?

A quantization table is often used in DCT-based image/video compression, where each frequency location is usually assigned an individual quantization step size to uniformly quantize the DCT coefficients at that frequency. Quantization table design is often seen as a rate-allocation problem, which is important to the RD performance of a DCT-based image/video compression system. The design of a quantization table has been well studied in traditional hard-decision quantization (HDQ) settings [19], [62], [42].

Recently, a more advanced quantization technique called soft-decision quantization (SDQ) has been developed for image/video compression [79], [73], [74], [72], [70], [71], [16], [64], [22]. Because of its superiority over HDQ, SDQ or its suboptimal version called rate distortion optimized quantization (RDOQ) [22] has been well adopted in both video coding standards H.264/AVC [31] and HEVC [32]. In light of the increasing importance of SDQ in image/video compression, we are motivated to revisit quantization table design in the context of SDQ, as we raise the question–how to efficiently design a quantization table for image/video compression given a statistical model of DCT coefficients, where SDQ is considered?

3) With a statistical model and a quantization table, how can we efficiently design

quantizers and entropy coding for DCT-based image compression?

For DCT-based image compression, knowledge of the distribution of DCT coefficients is important to quantizers and entropy coding design, as the resulting statistical model can guide quantizer design and thus affect the subsequent entropy coding design. In addition, the quantizer design also depends on the quantization table design scheme when SDQ is considered (see Chapter 3 and Chapter 4 for details). To further improve the RD performance of DCT-based image compression, we raise the question–how to efficiently design quantizers and entropy coding, given a new statistical model and a quantization table design scheme considering SDQ.

1.2 Thesis contributions

The purpose of this thesis is mainly to answer the three questions raised in the last section one by one:

1) In answering the first question in Section 1.1, a further study on the distribution of DCT coefficients is conducted. To better handle the heavy tail phenomenon commonly seen in DCT coefficients, a new model dubbed transparent composite model (TCM) is developed and justified. For continuous DCT coefficient sources, a TCM with the parametric distribution being truncated Laplacian distribution, i.e., Laplacian TCM (LPTCM) was first proposed in [75], which is extended to the discrete case in this thesis; for discrete DCT coefficient sources, a TCM with the parametric distribution being truncated geometric distribution, i.e., geometric TCM (GMTCM) is proposed. As shown in our goodness of fitting tests, the LPTCM and GMTCM achieve superior fitting accuracy in modelling DCT coefficients. Together with an additional capability of nonlinear data reduction while maintaining the modelling simplicity and practicality, the LPTCM and GMTCM can be

efficiently applied online to guide the lossy algorithm design for DCT-based image/video compression.

2) in answering the second question in Section 1.1, quantization table design is revisited for image/video compression where SDQ is considered. An optimal distortion profile scheme (OptD) for quantization table design is proposed in this thesis, which can be easily performed based on a given statistical model with negligible computational complexity. Unlike conventional approaches where quantization table design is bundled with a specific encoding method, the OptD assumes optimal SDQ encoding and design a quantization table for the purpose of reconstruction. Interestingly, we convert the traditional rate-allocation problem into a distortion allocation problem, and thus the resulting OptD can be efficiently applied to any image/video compression systems where a quantization table is involved, to improve the RD performance (in terms of rate and distortion tradeoff). Based on the OptD, we further propose a model-based quantization table optimization algorithm to boost the RD performance for JPEG type encoding.

3) in answering the third question in Section 1.1, an efficient non-predictive DCT-based image compression system is developed based on the LPTCM and OptD, where the quantizers and entropy coding are completely re-designed. A new SDQ algorithm is also proposed to jointly optimize the quantization and entropy coding. As shown in our subjective and objective tests, the proposed image compression system provides high coding efficiency both in terms of rate vs peak signal-to-noise ratio (PSNR) and rate vs visual quality, with low computational complexity, while having an additional multiresolution capability, which make it a good alternative for real-time image processing applications.

1.3 Thesis organization

The rest of the thesis is organized as follows. Chapter 2 proposes the TCM with justifications including the LPTCM and GMTCM, with the data reduction capability shown in the end of the chapter. Chapter 3 presents the OptD for quantization table design in the context of SDQ, which is applied to JPEG encoding and other image coding systems based on the Laplacian model. Chapter 4 discusses the new DCT-based image compression system based on the LPTCM and OptD, with both subjective and objective test results and its multiresolution feature demonstrated. Chapter 5 summarizes the thesis and proposes relative future work.

Chapter 2

Transparent Composite Model for DCT coefficients

This chapter proposes a new statistical model called TCM, which is separated into two cases for continuous DCT coefficient sources, i.e., continuous TCM, and discrete DCT coefficient sources, i.e., discrete TCM. Section 2.1 reviews various statistical models proposed in the literature and the relative measurement for modelling accuracy. Section 2.2 demonstrates the heavy tail phenomenon in DCT coefficients. The continuous TCM and discrete TCM are presented in Section 2.3 and Section 2.4, respectively. Section 2.5 shows experimental results on modelling accuracy by TCMs, followed by discussions on the data reduction capability of the TCM in Section 2.6. Finally, we summarize this chapter in Section 2.7.

2.1 Literature review

2.1.1 Models in the literature for DCT coefficients

As discussed in Section 1.1, the popularity of DCT [7], [10] has motivated many research works on statistical models for DCT coefficients, mainly including Gaussian [6], generalized Gaussian [33], [25], Laplacian [43], [36], [25], [50], and Cauchy [21] distributions. We review each of them in the following.

Gaussian distributions

Gaussian distributions are widely used for modelling DCT coefficients, and its justification roots in the central limit theorem (CLT). The probability density function (pdf) of Gaussian distribution with a zero mean for modelling DCT data is given by

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-|y|^2/2\sigma^2}, \tag{2.1}$$

where σ is the standard deviation, and the variance is therefore σ^2. A comprehensive collection of distributions based on Gaussian pdf were studied in [6]. However, it was observed that DCT coefficients for natural images/video usually possess a tail heavier than Gaussian distributions [33]. Consequently, generalized Gaussian distributions have been suggested for modelling DCT coefficients.

Generalized Gaussian distributions

The pdf of generalized Gaussian distribution (GGD) with a zero mean for modelling DCT data is as follows,

$$f(y) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|y|/\alpha)^\beta} \tag{2.2}$$

where α is a positive scale parameter, β defines a positive shape parameter, and $\Gamma(\cdot)$ denotes the gamma function.

It is easy to see that when $\beta = 1$, the GGD is de-generalized to a Laplacian distribution. When $\beta = 2$, it becomes the Gaussian distribution with variance $\alpha^2/2$. With the free choice of the scale parameter α and the shape parameter β, the GGD has shown an effective way to parameterize a family of symmetric distributions spanning from Gaussian to uniform distributions, and a family of symmetric distributions spanning from Laplacian to Gaussian distributions. As mentioned above, DCT coefficient distributions are observed to posses heavy tails. In this regard, the GGD allows for either heavier-than-Gaussian tails with $\beta < 2$, heavier-than-Laplacian tails with $\beta < 1$, or lighter-than-Gaussian tails with $\beta > 2$. As such, the GG model outperforms in general both the Gaussian and Laplacian models in terms of modelling accuracy for modelling DCT coefficients. Nevertheless, the benefit of accurate modelling by the GG model comes with some inevitable drawbacks–its lack of closed-form cumulative distribution function (cdf) and high complexity for parameter estimation. As shown in [33], given a sequence of samples $Y_i, i = 1, \cdots, n$, the ML estimation for β is to solve the following equation,

$$\frac{\psi(1/\beta + 1) + \log(\beta)}{\beta^2} + \frac{1}{\beta^2} \log(\frac{1}{n} \sum_{i=1}^{n} |Y_i|^\beta) - \frac{\sum_{i=1}^{n} |Y_i|^\beta \log |Y_i|}{\beta \sum_{i=1}^{n} |Y_i|^\beta} = 0, \tag{2.3}$$

where

$$\psi(\tau) = \gamma + \int_0^1 (1 - t^{\tau-1})(1 - t)^{-1} dt$$

and $\gamma = 0.577...$ denotes the Euler constant. Clearly, the terms $\sum_{i=1}^{n} |Y_i|^\beta \log |Y_i|$ and $\beta \sum_{i=1}^{n} |Y_i|^\beta$ yield a significant amount of computation when a numerical iterative solution of β is used.

Laplacian distributions

Due to its simplicity and fair modelling performance, the Laplacian model becomes the most popular choice in use [43], [36], [25], [50], with its pdf given as follows,

$$f(y) = \frac{1}{2\lambda} e^{-(|y|/\lambda)}, \tag{2.4}$$

where λ denotes a positive scale parameter. Given a sequence of samples $Y_i, i = 1, \cdots, n$, the ML estimate of λ can be easily computed as

$$\lambda = \frac{1}{n} \sum_{i=1}^{n} |Y_i|. \tag{2.5}$$

Cauchy distributions

Cauchy distribution was inspired by the heavy-tail observations of DCT coefficients, with the following pdf

$$f(y) = \frac{r}{\pi} \frac{1}{(y - y_0)^2 + r^2}, \tag{2.6}$$

where y_0 is a location parameter and r stands for a scale parameter. Our studies on comparing the Cauchy model with the GGD show that the GGD generally provides a better goodness of fitting than the Cauchy model. In addition, the application of the Cauchy distribution is also limited due to the fact that it does not have finite moments of any order, causing difficulties for its parameter estimation.

2.1.2 Measurement for modelling accuracy

There are three methods commonly used in the literature for testing modelling accuracy, i.e., Kolmogorov-Smirnov (KS) test, Kullback-Leibler (KL)[44] divergence, and χ^2 test[33].

In general, the KS test is more sensitive to the main portion than to the tail part. The χ^2 test, on the other hand, shifts its focus to the tail portion more than the KS test does. The KL divergence, as shown by its use of logarithm, stands in the middle of KS and χ^2 test in terms of balancing between the fitness of the main portion and the fitness of the tail part.

Similar as in [33], this thesis prefers the χ^2 test over the KS test for measuring the modelling accuracy. Besides the justification provided in [33] for using the χ^2 test rather than KS test as χ^2 gives a more meaningful guidance to source coding, our preference also roots in the heavy-tail phenomena of DCT coefficients. Specifically, χ^2 test better characterizes a statistically insignificant tail portion in a distribution while the KS test tends to overlook the tail part. In the following, more detailed discussions are present for the heavy tail phenomena. Besides using χ^2 test, we also use the KL divergence for comparing modelling accuracy, due to its balance between the emphasis on the main portion by the KS test and the emphasis on the tail by the χ^2 test.

Given a sequence of sample probabilities $\{p_i\}$, and a sequence of model probabilities $\{q_i\}$, the KL divergence of the model from the observations is

$$KL = \sum_i p_i \cdot \ln \frac{p_i}{q_i}, \tag{2.7}$$

where $0 \ln 0$ is defined as 0. The χ^2 test is defined as

$$\chi^2 = \sum_i \frac{n \cdot (p_i - q_i)^2}{q_i}, \tag{2.8}$$

where n is the total number of samples.

2.2 Heavy tail observations in DCT coefficients

Gaussian, GG, and Laplacian distributions all decay exponentially fast. As illustrated in Figure 2.1 (in all histograms of this thesis, the DCT coefficient intervals are set to be 1), however, DCT coefficients usually possess a much heavier tail. Figure 2.1 was obtained by applying the floating-point 8×8 DCT to the well-known 512×512 Lena image (AC2 represents the first AC frequency in the zigzag scanning order), where the yellow bars show the histogram of the DCT coefficients. It is evident from the left panel of Figure 2.1 that the histogram of the DCT coefficients first decays quite rapidly for the main portion of DCT coefficients and then becomes relatively flat for the tail portion of DCT coefficients.

The right panel of Figure 2.1 zooms in the tail portion and further compares the histogram of DCT coefficients against the GG and Laplacian models, where the yellow bars again represent the histogram of DCT coefficients, and the red and black curves show results from the GG and Laplacian models, respectively. In Figure 2.1, the ML estimates of the parameters of the GG model were computed via Matlab codes from [15] while the λ value of the Laplacian model was computed using (2.5). For both models, the χ^2 tests were performed to evaluate their respective modelling accuracy. According to the χ^2 test, the GG model significantly outperforms the Laplacian model. Furthermore, in Figure 2.1, the obtained shape parameter β is much smaller than 1, meaning that the resulting GG distribution possesses a tail heavier than that of the Laplacian distribution. In comparison with the real data histogram shown in Figure 2.1, however, the GG model still suffers from an exponentially bounded tail, which is much lighter than that of the DCT coefficients.

The heavy tail phenomenon in the Lena image is widely observed in other images as well. As shown in [33], the estimated shape parameter β for the GG distribution for various images is less than 1 in most cases, indicating that the data distribution possesses a tail

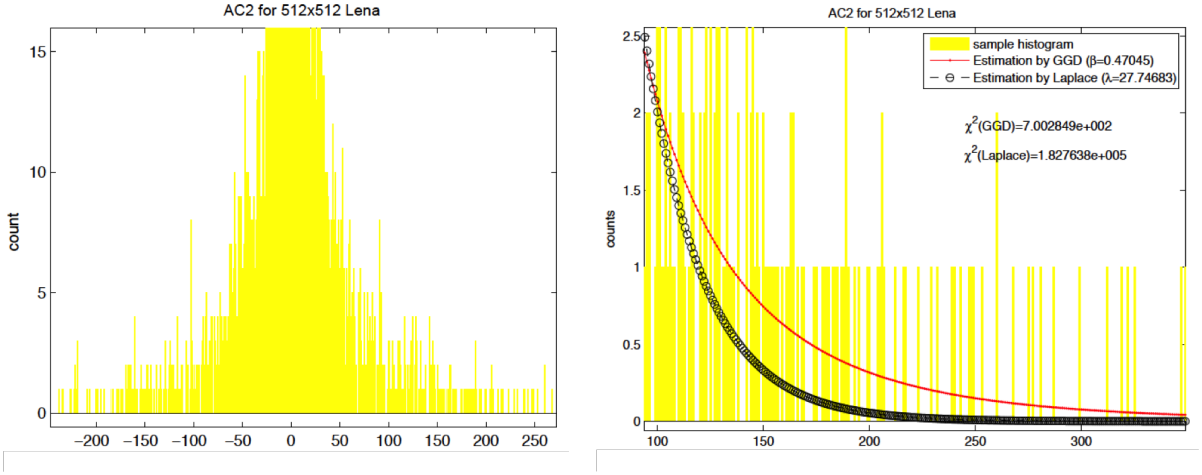

Figure 2.1: Histogram and the tail of an AC component in the 8×8 DCT block of Lena.

heavier than that of the Laplacian distribution. In [21], it was also observed that the tail of DCT coefficients in video coding is much heavier than that of the Laplacian distribution, and a Cauchy distribution was used instead for deriving rate and distortion models for DCT coefficients. However, as mentioned before, the Cauchy model does not model the main portion of DCT coefficients effectively, and is in general inferior to the GG model in term of the overall modelling accuracy. Therefore, it is desirable to have a new model which can balance well the main portion and tail portion of DCT coefficients while having both simplicity and superior modelling accuracy.

2.3 Continuous Transparent Composite Model

To better handle the heavy tail in DCT data, we now separate the tail of DCT coefficients from the main part and use a different model to model each of them. Since DCT coefficients in the tail portion are insignificant statistically, each of them often appears once or a few times. Hence it would make sense to model them separately by a uniform distribu-

tion while modelling the main portion by a parametric distribution, yielding a model we call a transparent composite model (TCM). This *composite* model introduces a boundary parameter to control which model to use for any given DCT coefficient; it is marked as *transparent* because there is no ambiguity regarding which model (the uniform or parametric) a given DCT coefficient will fall into once the TCM is determined. In this section, we assume that DCT data are continuous and consider continuous TCMs.

2.3.1 Description of general continuous TCMs

Consider a pdf $f(y|\theta)$ with parameters $\theta \in \Theta$, where θ could be a vector, and Θ is the parameter space, and denote by $F(y|\theta)$ the corresponding cdf. Assume that $f(y|\theta)$ is symmetric in y with respect to the origin, and $F(y|\theta)$ is concave as a function of y in the region $y \geq 0$. It is easy to verify that Laplacian, Gaussian, GG, and Cauchy distributions all satisfy this assumption. The TCM based on $F(y|\theta)$ is defined as

$$p(y|Y_c, b, \theta) \triangleq \begin{cases} \frac{b}{2F(Y_c|\theta)-1} f(y|\theta), & |y| < Y_c \\ \frac{1-b}{2(A-Y_c)}, & Y_c < |y| \leq A \\ \max\{\frac{b}{2F(Y_c|\theta)-1} f(Y_c|\theta), \frac{1-b}{2(A-Y_c)}\}, & |y| = Y_c \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

where $0 \leq b \leq 1$, $0 < Y \leq Y_c < A$ denotes truncation point or separation boundary, which separates the tail of DCT coefficients from the main body part, and A represents the largest magnitude a sample y can take. Here both A and Y are assumed to be known. In addition, the uniform distribution $\frac{1}{2(A-Y_c)}$ will be called the outlier distribution, and samples from the outlier distribution will be referred to as outliers. Accordingly, samples from the parametric distribution will be referred to as inliers.

2.3.2 ML estimate of TCM parameters

Let $Y_1^n = Y_1, Y_2, \cdots, Y_n$ be a sequence of DCT coefficients. Assume that Y_1^n behaves according to the TCM defined in (2.9) with $Y_{max} \triangleq \max\{|Y_i| : 1 \le i \le n\} < A$ and $Y_{max} \ge Y$. (When $Y_{max} < Y$, there are no outliers and the ML estimate of Y_c and b is equal to Y and 1, respectively.) We next show how to compute the ML estimate of Y_c, b and θ.

Given Y_1^n with $Y \le Y_{max} < A$, let

$$\begin{aligned} N_1(Y_c) &\triangleq \{i : |Y_i| < Y_c\} \\ N_2(Y_c) &\triangleq \{i : Y_c < |Y_i|\} \\ N_3(Y_c) &\triangleq \{i : |Y_i| = Y_c\}. \end{aligned}$$

Then the log-likelihood function $g(Y_c, b, \theta|Y_1^n)$ according to (2.9) is equal to

$$\begin{aligned} & g(Y_c, b, \theta|Y_1^n) \\ & \stackrel{1)}{=} |N_2(Y_c)| \ln(1-b) + |N_1(Y_c)| \ln b + \sum_{i \in N_1(Y_c)} \ln f(Y_i|\theta) \\ & + |N_3(Y_c)| \max\{\ln \frac{1-b}{2(A-Y_c)}, \ln \frac{b f(Y_c|\theta)}{[2F(Y_c|\theta)-1]}\} \\ & - |N_2(Y_c)| \ln 2(A-Y_c) - |N_1(Y_c)| \ln [2F(Y_c|\theta)-1] \end{aligned} \tag{2.10}$$

where $|S|$ denotes the cardinality of a finite set S, and the equality 1) is due to (2.9) and the fact that $\ln z$ is strictly increasing in the region $z > 0$. Since $F(y|\theta)$ is nondecreasing with respect to y, we have $g(Y_c, b, \theta|Y_1^n) \le g(Y_{max}, b, \theta|Y_1^n)$ for any $Y_{max} < Y_c < A$, which leads to:

$$\begin{aligned} & \max\{g(Y_c, b, \theta|Y_1^n) : Y \le Y_c < A, 0 \le b \le 1, \theta\} \\ & = \max\{g(Y_c, b, \theta|Y_1^n) : Y \le Y_c \le Y_{max}, 0 \le b \le 1, \theta\}. \end{aligned} \tag{2.11}$$

To continue, we now sort $|Y_1|, |Y_2|, \cdots, |Y_n|$ in ascending order into $W_1 \le W_2 \le \cdots \le W_n$. Note that $W_n = Y_{max}$. Let m be the smallest integer i such that $W_i \ge Y$. Define $I_m = (Y, W_m)$ and for any $m < i \le n$, $I_i = (W_{i-1}, W_i)$. Then it is easy to see that the interval $[Y, Y_{max}]$ can be decomposed as

$$[Y, Y_{max}] = \{Y, W_m, W_{m+1}, \cdots, W_n\} \cup (\cup_{i=m}^{n} I_i)$$

which, together with (2.11), implies that

$$\begin{aligned}
&\max\{g(Y_c, b, \theta|Y_1^n) : Y \le Y_c < A, 0 \le b \le 1, \theta\} \\
&= \max_{0\le b\le 1} \max_{\theta} \max_{Y_c \in [Y, Y_{max}]} g(Y_c, b, \theta|Y_1^n) \\
&= \max_{b,\theta} \max\{g(Y, b, \theta|Y_1^n), g(W_i, b, \theta|Y_1^n), \\
&\qquad \sup_{Y_c \in I_i} [g(Y_c, b, \theta|Y_1^n)] : m \le i \le n\}. \qquad (2.12)
\end{aligned}$$

Note that for any nonempty I_i with $i > m$, $N_1(Y_c)$ and $N_2(Y_c)$ remain the same and $N_3(Y_c)$ is empty for all $Y_c \in I_i$. Since by assumption $F(y|\theta)$ as a function of y is concave, it is not hard to verify that as a function of Y_c, $(-|N_2(Y_c)| \ln 2(A - Y_c) - |N_1(Y_c)| \ln [2F(Y_c|\theta) - 1])$ is convex over $Y_c \in I_i$, and hence its value over $Y_c \in I_i$ is upper bounded by the maximum of its value at $Y_c = W_i$ and $Y_c = W_{i-1}$, i.e., the endpoints of I_i. Therefore, in view of (2.11), we have

$$\sup_{Y_c \in I_i} [g(Y_c, b, \theta|Y_1^n)] \le \max\{g(W_{i-1}, b, \theta|Y_1^n), g(W_i, b, \theta|Y_1^n)\}. \qquad (2.13)$$

When I_m is nonempty, a similar argument leads to

$$\sup_{Y_c \in I_m} [g(Y_c, b, \theta|Y_1^n)] \le \max\{g(Y, b, \theta|Y_1^n), g(W_m, b, \theta|Y_1^n)\}. \qquad (2.14)$$

Putting (2.12) to (2.14) together yields

$$\begin{aligned}
&\max\{g(Y_c, b, \theta|Y_1^n) : Y \le Y_c < A, 0 \le b \le 1, \theta\} = \\
&\max_{b,\theta} \max\{g(Y, b, \theta|Y_1^n), g(W_i, b, \theta|Y_1^n) : m \le i \le n\}. \qquad (2.15)
\end{aligned}$$

Therefore, the ML estimate of Y_c is equal to one of $Y, W_m, W_{m+1}, \cdots, W_n$.

We are now led to investigating $\max_{b,\theta} g(Y_c, b, \theta|Y_1^n)$ for each $Y_c \in \{Y, W_m, W_{m+1}, \cdots, W_n\}$. Define

$$N_1^+(Y_c) \triangleq \{i : |Y_i| \leq Y_c\} \text{ and } N_2^+(Y_c) \triangleq \{i : Y_c \leq |Y_i|\}.$$

Further define:

$$\begin{aligned} g^+(Y_c, b, \theta|Y_1^n) \triangleq & (|N_2(Y_c)|) \left[\ln(1-b) - \ln 2(A - Y_c)\right] \\ & + |N_1^+(Y_c)| \ln \frac{b}{2F(Y_c|\theta) - 1} + \sum_{i \in N_1^+(Y_c)} \ln f(Y_i|\theta) \end{aligned} \tag{2.16}$$

$$\begin{aligned} g^-(Y_c, b, \theta|Y_1^n) \triangleq & \left(|N_2^+(Y_c)|\right) \left[\ln(1-b) - \ln 2(A - Y_c)\right] \\ & + |N_1(Y_c)| \ln \frac{b}{2F(Y_c|\theta) - 1} + \sum_{i \in N_1(Y_c)} \ln f(Y_i|\theta) \end{aligned} \tag{2.17}$$

Note that the difference between $g^+(Y_c, b, \theta|Y_1^n)$ and $g^-(Y_c, b, \theta|Y_1^n)$ lies in whether or not we regard Y_c itself as an outlier when Y_c is equal to some W_i. Comparing (2.11) with (2.16) and (2.17), we have

$$g(Y_c, b, \theta|Y_1^n) = \max\{g^+(Y_c, b, \theta|Y_1^n), g^-(Y_c, b, \theta|Y_1^n)\} \tag{2.18}$$

and hence

$$\begin{aligned} & \max_{b,\theta} g(Y_c, b, \theta|Y_1^n) = \\ & \quad \max\{\max_{b,\theta} g^+(Y_c, b, \theta|Y_1^n), \max_{b,\theta} g^-(Y_c, b, \theta|Y_1^n)\}. \end{aligned} \tag{2.19}$$

$$\begin{aligned} \text{Let} \quad (b(Y_c), \theta(Y_c)) &\triangleq \arg\max_{b,\theta} g(Y_c, b, \theta|Y_1^n) \\ (b^+(Y_c), \theta^+(Y_c)) &\triangleq \arg\max_{b,\theta} g^+(Y_c, b, \theta|Y_1^n) \\ (b^-(Y_c), \theta^-(Y_c)) &\triangleq \arg\max_{b,\theta} g^-(Y_c, b, \theta|Y_1^n). \end{aligned}$$

Then from (2.16) and (2.17), it is not hard to see that

$$b^+(Y_c) = \frac{|N_1^+(Y_c)|}{n} \text{ and } b^-(Y_c) = \frac{|N_1(Y_c)|}{n} \tag{2.20}$$

and $\theta^+(Y_c)$ and $\theta^-(Y_c)$ are the ML estimate of θ for the truncated distribution $\frac{1}{2F(Y_c|\theta)-1}f(y|\theta)$ over the sample sets $\{Y_i : i \in N_1^+(Y_c)\}$ and $\{Y_i : i \in N_1(Y_c)\}$, respectively. In view of (2.19), one can then determine $(b(Y_c), \theta(Y_c))$ by setting

$$(b(Y_c), \theta(Y_c)) = \begin{cases} (b^+(Y_c), \theta^+(Y_c)) & \text{if } (c) \\ (b^-(Y_c), \theta^-(Y_c)) & \text{otherwise} \end{cases} \tag{2.21}$$

where (c) stands for

$$g^+(Y_c, b^+(Y_c), \theta^+(Y_c)|Y_1^n) \geq g^-(Y_c, b^-(Y_c), \theta^-(Y_c)|Y_1^n).$$

Finally, the ML estimate of (Y_c, b, θ) can be determined as

$$\begin{aligned} Y_c^* &= \underset{Y_c \in \{d, W_m, \cdots, W_n\}}{\arg\max} g(Y_c, b(Y_c), \theta(Y_c)|Y_1^n) \\ b^* &= b(Y_c^*) \\ \theta^* &= \theta(Y_c^*). \end{aligned} \tag{2.22}$$

Summarizing the above derivations into Algorithm 1 for computing (Y_c^*, b^*, θ^*), we have proved the following result.

Theorem 1 *The vector (Y_c^*, b^*, θ^*) computed by Algorithm 1 is indeed the ML estimate of (Y_c, b, θ) in the TCM specified in (2.9).*

Depending on whether or not Step 6 in Algorithm 1 can be implemented efficiently, the computation complexity of Algorithm 1 varies from one parametric family $f(y|\theta)$ to another. For some parametric family $f(y|\theta)$ such as Laplacian distributions, Step 6 can be

Algorithm 1 A general algorithm for estimating (Y_c, b, θ)

Sort $\{|Y_i|\}_{i=1}^n$ in ascending order into $W_1 \leq \cdots \leq W_n$.

Determine $m = \min\{i : W_i \geq d\}$.

for each $Y_c \in \{d, W_m, W_{m+1}, \cdots, W_n\}$ **do**

  Set $N_1^+(Y_c) = \{i : |Y_i| \leq Y_c\}, N_1(Y_c) = \{i : |Y_i| < Y_c\}$.

  Compute $b^+(Y_c) = \frac{|N_1^+(Y_c)|}{n}$ and $b^-(Y_c) = \frac{|N_1(Y_c)|}{n}$.

  Determine $\theta^+(Y_c)$ and $\theta^-(Y_c)$ to be the ML estimate of θ for the truncated distribution $\frac{f(y|\theta)}{2F(Y_c|\theta)-1)}$ over $\{Y_i : i \in N_1^+(Y_c)\}$ and $\{Y_i : i \in N_1(Y_c)\}$, respectively.

  if $g^+(Y_c, b^+(Y_c), \theta^+(Y_c)|Y_1^n) \geq g^-(Y_c, b^-(Y_c), \theta^-(Y_c)|Y_1^n)$ **then**

    set $(b(Y_c), \theta(Y_c)) = (b^+(Y_c), \theta^+(Y_c))$

  else

    set $(b(Y_c), \theta(Y_c)) = (b^-(Y_c), \theta^-(Y_c))$.

  end if

end for

Determine $Y_c^* = \arg\max_{Y_c \in \{d, W_m, \cdots, W_n\}} g(Y_c, b(Y_c), \theta(Y_c)|Y_1^n)$.

Set $b^* = b(Y_c^*)$ and $\theta^* = \theta(Y_c^*)$.

easily solved and hence Algorithm 1 can be implemented efficiently. On the other hand, when $f(y|\theta)$ is the GG family, Step 6 is quite involved. In the next two subsections, we will examine Step 6 in two cases: (1) $f(y|\theta)$ is the Laplacian family, and the corresponding TCM is referred to as the LPTCM; and (2) $f(y|\theta)$ is the GG family, and the corresponding TCM is referred to as the GGTCM.

2.3.3 LPTCM

Plugging the Laplacian density function in (2.4) into (2.9), we get the LPTCM given by

$$p(y|Y_c,b,\lambda) \triangleq \begin{cases} \frac{b}{1-e^{-Y_c/\lambda}}\frac{1}{2\lambda}e^{-|y|/\lambda} & \text{if } |y|<Y_c \\ \frac{1-b}{2(A-Y_c)} & \text{if } Y_c<|y|\le A \\ \max\{\frac{b}{1-e^{-Y_c/\lambda}}\frac{1}{2\lambda}e^{-|y|/\lambda}, \frac{1-b}{2(A-Y_c)}\} & \text{if } |y|=Y_c \\ 0 & \text{otherwise.} \end{cases} \tag{2.23}$$

With reference to Step 6 in Algorithm 1, let S be either $N_1^+(Y_c)$ or $N_1(Y_c)$. Then Step 6 in Algorithm 1 is equivalent to determining the ML estimate (denoted by λ_{Y_c}) of λ in the truncated Laplacian distribution

$$p(y|\lambda) \triangleq \begin{cases} \frac{1}{1-e^{-Y_c/\lambda}}\frac{1}{2\lambda}e^{-|y|/\lambda} & \text{if } |y|\le Y_c \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

from the sample set $\{Y_i : i\in S\}$. Since $|Y_i|\le Y_c$ for any $i\in S$, the log-likelihood function of the sample set $\{Y_i : i\in S\}$ with respect to $p(y|\lambda)$ is equal to

$$L(\lambda) \triangleq -|S|[\ln 2\lambda + \ln(1-e^{-Y_c/\lambda})] - \frac{1}{\lambda}\sum_{i\in S}|Y_i|.$$

Then we have $\lambda_{Y_c} = \arg\max_{0\le\lambda\le\infty} L(\lambda)$. It is not hard to verify that $L(1/t)$ as a function of $t>0$ is strictly concave. Computing the derivative of $L(\lambda)$ with respect to λ and setting

it to 0 yields

$$\lambda - \frac{Y_c \cdot e^{-Y_c/\lambda}}{1 - e^{-Y_c/\lambda}} - \frac{1}{|S|} \sum_{i \in S} |Y_i| = 0. \tag{2.25}$$

It can be shown (see the proof of Theorem 2 below) that $s(\lambda) \triangleq \lambda - \frac{Y_c \cdot e^{-Y_c/\lambda}}{1 - e^{-Y_c/\lambda}}$ is a strictly increasing function of $\lambda > 0$, and $\lim_{\lambda \to 0^+} s(\lambda) = 0$ and $\lim_{\lambda \to \infty} = \frac{Y_c}{2}$. Let

$$C = \frac{1}{|S|} \sum_{i \in S} |Y_i|. \tag{2.26}$$

Then it follows: (1) when $C = 0$, $\lambda_{Y_c} = 0$, in which case the corresponding truncated Laplacian distribution is de-generated to a delta function; (2) when $C \geq Y_c/2$, $\lambda_{Y_c} = \infty$, in which case the corresponding truncated Laplacian distribution is de-generated to the uniform distribution over $[-Y_c, Y_c]$, and (3) when $0 < C < Y_c/2$, λ_{Y_c} is equal to the unique root to (2.25).

We are now led to solving (2.25) when $0 < C < Y_c/2$, for which Algorithm 2 is proposed. It can be shown Algorithm 2 converges exponentially fast, as by Theorem 2.

Algorithm 2 Estimating λ for a truncated Laplacian model

1: Compute $C = \frac{1}{|S|} \sum_{i \in S} |Y_i|$. Set $\lambda_{Y_c} = 0$ if $C = 0$; or set $\lambda_{Y_c} = \infty$ if $C \geq Y_c/2$.

2: Initialization: set $\lambda_0 = C$;

3: For $i \geq 1$, compute

$$\lambda_i = C + \frac{Y_c \cdot e^{-Y_c/\lambda_{i-1}}}{1 - e^{-Y_c/\lambda_{i-1}}}; \tag{2.27}$$

4: Repeat Step 3 until $\lambda_i - \lambda_{i-1} < \epsilon$, where $\epsilon > 0$ is a small prescribed threshold.

Theorem 2 *Assume that $0 < C < Y_c/2$. Then λ_i computed in Step 3 of Algorithm 2 strictly increases and converges exponentially fast to λ_{Y_c} as $i \to \infty$.*

Proof: Define $r(\lambda) = \lambda - \frac{Y_c \cdot e^{-Y_c/\lambda}}{1-e^{-Y_c/\lambda}} - C$. It is not hard to verify that the derivative of $r(\lambda)$ with respect to λ is

$$r'(\lambda) = 1 - \frac{e^{-Y_c/\lambda}}{[1-e^{-Y_c/\lambda}]^2}\frac{Y_c^2}{\lambda^2} > 0 \tag{2.28}$$

for any $\lambda > 0$. Thus, $r(\lambda)$ is strictly increasing over $\lambda > 0$.

Since $\lambda_0 = C > 0$, it follows from (2.27) that $\lambda_1 > \lambda_0$. In general, for any $i \geq 1$, we have

$$\begin{aligned} \lambda_{i+1} - \lambda_i &= \frac{Y_c \cdot e^{-Y_c/\lambda_i}}{1-e^{-Y_c/\lambda_i}} - \frac{Y_c \cdot e^{-Y_c/\lambda_{i-1}}}{1-e^{-Y_c/\lambda_{i-1}}} \\ &= Y_c\left[\frac{1}{e^{Y_c/\lambda_i}-1} - \frac{1}{e^{Y_c/\lambda_{i-1}}-1}\right] \end{aligned} \tag{2.29}$$

which implies that $\lambda_{i+1} - \lambda_i > 0$ whenever $\lambda_i - \lambda_{i-1} > 0$. By mathematic induction, it then follows that λ_i strictly increases as i increases.

We next show that all λ_i, $i \geq 1$, are bounded. Indeed, it follows from (2.27) that

$$\begin{aligned} r(\lambda_i) &= \lambda_i - \frac{Y_c \cdot e^{-Y_c/\lambda_i}}{1-e^{-Y_c/\lambda_i}} - C \\ &= \lambda_i - \lambda_{i+1} \\ &< 0 \end{aligned}$$

which, together with (2.28) and the fact that $r(\lambda_{Y_c}) = 0$, implies that $\lambda_i < \lambda_{Y_c}$. Therefore λ_i converges as $i \to \infty$. Letting $i \to \infty$ in (2.27) yields

$$\lim_{i\to\infty} \lambda_i = \lambda_{Y_c}. \tag{2.30}$$

All remaining is to show that the convergence speed in (2.30) is exponentially fast. To this end, let $\delta \triangleq \max_{\lambda_0 \leq \lambda \leq \lambda_{Y_c}} \frac{e^{-Y_c/\lambda}}{\left[1-e^{-Y_c/\lambda}\right]^2}\left(\frac{Y_c}{\lambda}\right)^2$. Then it follows from (2.28) that $\delta < 1$. This, together with (2.29), implies that $\lambda_{i+1} - \lambda_i \leq \delta(\lambda_i - \lambda_{i-1})$ for any $i \geq 1$, and hence λ_i converges to λ_{Y_c} exponentially fast. This completes the proof of Theorem 2.

Plugging Algorithm 2 into Step 6 in Algorithm 1, one then gets an efficient algorithm for computing the ML estimate of (Y_c, b, λ) in the LPTCM. To illustrate the effectiveness of the LPTCM, the resulting algorithm was applied to the same DCT coefficients shown in Figure 2.1. Figure 2.2 shows the resulting LPTCM against the histogram of DCT coefficients on the whole in each respective case. From Figure 2.2, it is clear that the LPTCM fits the histogram of DCT coefficients quite well and greatly improves upon the Laplacian model in each case. In comparison with the Laplacian model, it fits both the main and tail portions better. In terms of χ^2 values, it matches up to the GG model. More detailed comparisons will be presented in Section 2.5.

2.3.4 GGTCM

Plugging the GG density function in (2.2) into (2.9), we get the GGTCM given by

$$p(y|Y_c, b, \alpha, \beta) \triangleq \tag{2.31}$$
$$\begin{cases} \frac{b\beta}{2\alpha\gamma(1/\beta,(Y_c/\alpha)^\beta)} e^{-(|y|/\alpha)^\beta} & |y| < Y_c \\ \frac{1-b}{2(A-Y_c)} & Y_c < |y| \leq A \\ \max\{\frac{b\beta}{2\alpha\gamma(1/\beta,(Y_c/\alpha)^\beta)} e^{-(|y|/\alpha)^\beta}, \frac{1-b}{2(A-Y_c)}\} & |y| = Y_c \\ 0 & \text{otherwise} \end{cases}$$

where $\gamma(s, x)$ is defined as $\gamma(s, x) \triangleq \int_0^x t^{s-1}e^{-t}dt$. With reference to Algorithm 1, in this case, Step 6 in Algorithm 1 is equivalent to determining the ML estimate (denoted by $(\alpha_{Y_c}, \beta_{Y_c})$) of (α, β) in the truncated GG distribution

$$p(y|\alpha, \beta) \triangleq \begin{cases} \frac{\beta}{2\alpha\gamma(1/\beta,(Y_c/\alpha)^\beta)} e^{-(|y|/\alpha)^\beta} & \text{if } |y| \leq Y_c \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

from the sample set $\{Y_i : i \in S\}$. Since $|Y_i| \leq Y_c$ for any $i \in S$, the log-likelihood function of the sample set $\{Y_i : i \in S\}$ with respect to $p(y|\alpha, \beta)$ is equal to $L(\alpha, \beta) \triangleq$

$|S|\left[\ln\beta - \ln 2\alpha - \ln\gamma(\frac{1}{\beta}, (\frac{Y_c}{\alpha})^\beta)\right] - \sum_{i\in S}\left|\frac{Y_i}{\alpha}\right|^\beta$. Therefore $(\alpha_{Y_c}, \beta_{Y_c}) = \arg\max_{\alpha,\beta} L(\alpha, \beta)$. Computing the partial derivatives of $L(\alpha, \beta)$ with respect to α and β and setting them to zero yields

$$\begin{cases} \frac{1}{t} = \beta\left[\frac{1}{|S|}\sum_{i\in S}\left|\frac{Y_i}{Y_c}\right|^\beta + \frac{t^{1/\beta-1}e^{-t}}{\gamma(1/\beta,t)}\right] \\ \beta = \ln t - \frac{\int_0^t y^{1/\beta-1}e^{-y}\ln y dy}{\gamma(1/\beta,t)} + \frac{t\beta^2}{|S|}\sum_{i\in S}\left|\frac{Y_i}{Y_c}\right|^\beta \ln\frac{Y_i}{Y_c} \end{cases} \tag{2.33}$$

where $t = (Y_c/\alpha)^\beta$.

Unlike the case of LPTCM, however, solving (2.33) does not seem to be easy. In particular, at this point, we do not know whether (2.33) admits a unique solution. There is no developed algorithm with global convergence to compute such a solution either even if the solution is unique. As such, Step 6 in Algorithm 1 in the case of GGTCM is much more complicated than that in the case of LPTCM.

Suboptimal alternatives are to derive approximate solutions to (2.33). One approach is to solve the two equations in (2.33) iteratively. Together with this suboptimal solution to (2.33), Algorithm 1 was applied to to the same DCT coefficients shown in Figure 2.1. Figure 2.2 shows the resulting GGTCM against the histogram of DCT coefficients on the whole in each respective case. We note that the resulting GGTCM improves on the GG model marginally, which may be due to the suboptimal solution to (2.33).

2.4 Discrete Transparent Composite Model

In practice (particularly in lossy image and video coding), DCT is often designed and implemented as a mapping from an integer-valued space (e.g., 8-bits pixels) to another integer-valued space and gives rise to integer DCT coefficients. In addition, since most images and video are stored in a compressed format such as JPEG, H.264/AVC, etc., for

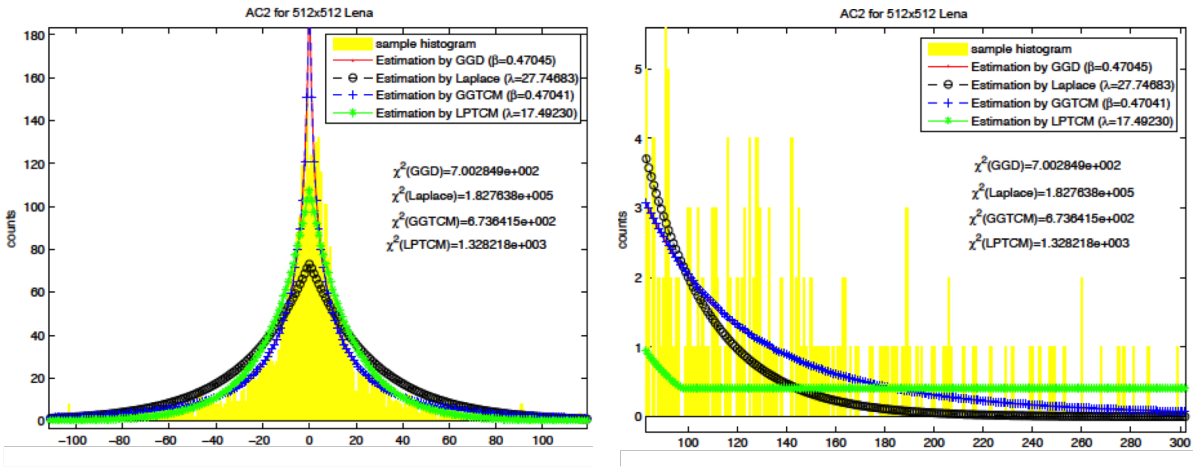

Figure 2.2: Illustration of the overall curves and tails of the LPTCM and GGTCM for an AC component in the 8×8 DCT block of Lena.

applications based on compressed images and video, DCT coefficients are available only in their quantized values. Therefore, it is desirable to establish a good model for discrete (integer or quantized) DCT coefficients as well.

This section proposes a discrete TCM. The particular discrete parametric distribution we will consider is a truncated geometric distribution, and the resulting discrete TCM is referred to as the GMTCM. To provide a uniform treatment for both integer and quantized DCT coefficients, we introduce a quantization factor of step size. Then both integer and quantized DCT coefficients can be regarded as integers multiplied by a properly chosen step size.

2.4.1 GMTCM

Uniform quantization with dead zone is widely used in image and video coding (see, for example, H.264/AVC and HEVC). Mathematically, a uniform quantizer with dead zone

Figure 2.3: Uniform quantization with deadzone.

and step size q is given by

$$Q(y) = q \times sign(y) \times round\left(\frac{||y| - u + q/2|}{q}\right), \tag{2.34}$$

where $q/2 \leq u < q$. Its input-output relationship is shown in Figure 2.3. Assume that the input y is distributed according to the Laplacian distribution in (2.4). Then the quantized index $sign(y) \times round\left(\frac{||y|-u+q/2|}{q}\right)$ is distributed as follows

$$\begin{aligned} p_0 &= 1 - e^{-\frac{u}{\lambda}} \\ p_i &= \frac{1}{2} e^{-\frac{u}{\lambda}} [1 - e^{-\frac{q}{\lambda}}] e^{-\frac{q}{\lambda}(|i|-1)}, i = \pm 1, \pm 2, \cdots \end{aligned} \tag{2.35}$$

With the help of q, discrete (integer or quantized) DCT coefficients then take values of integers multiplied by q. (Hereafter, these integers will be referred to as DCT indices.) Note that p_i in (2.35) is essentially a geometric distribution. Using a geometric distribution to model the main portion of discrete DCT coefficients, we then get the GMTCM given by

$$\begin{cases} p_0 = bp \\ p_i = \frac{b(1-p)}{2} e^{-\frac{q}{\lambda}(|i|-1)} \frac{1-e^{-q/\lambda}}{1-e^{-qK/\lambda}}, & 1 \leq |i| \leq K \\ p_i = \frac{1-b}{2(A-K)}, & K < |i| \leq A \end{cases} \tag{2.36}$$

where $0 \leq p \leq 1$ is the probability of the zero coefficient, $0 \leq b \leq 1$, $1 \leq K \leq A$ represents the truncation point, and A is the largest index in a given sequence of DCT indices. Here A is assumed known, and b, p, λ and K are model parameters.

2.4.2 ML Estimate of GMTCM parameters

Algorithms

Let $u^n = u_1, u_2, \cdots, u_n$ be a sequence of DCT indices. Assume that u^n behaves according to the GMTCM defined by (2.36) with $u_{max} \triangleq \max\{|u_i| : 1 \leq i \leq n\} \leq A$. We now investigate how to compute the ML estimate $(b^*, p^*, \lambda^*, K^*)$ of (b, p, λ, K) from u^n.

Let $N_0 = \{j : u_j = 0\}, N_1(K) = \{j : 0 < |u_j| \leq K\}$, and $N_2(K) = \{j : |u_j| > K\}$. The log-likelihood function of u^n according to (2.36) is equal to

$$\begin{aligned} & G(K, \lambda, b, p) \\ & \triangleq |N_2(K)| \ln(1-b) + (|N_0| + |N_1(K)|) \ln b \\ & + |N_0| \ln p + |N_1(K)| \ln(1-p) - |N_2(K)| \ln 2(A-K) \\ & + |N_1(K)| \ln \frac{1 - e^{-\frac{q}{\lambda}}}{2(1 - e^{-\frac{q}{\lambda}K})} - \frac{q}{\lambda} \sum_{j \in N_1(K)} (|u_j| - 1). \end{aligned} \tag{2.37}$$

Then we have

$$(b^*, p^*, \lambda^*, K^*) = \arg\max_{b,p,\lambda,K} G(K, \lambda, b, p). \tag{2.38}$$

For any K, let $L(K, \lambda) \triangleq |N_1(K)| \ln \frac{1-e^{-\frac{q}{\lambda}}}{2(1-e^{-\frac{q}{\lambda}K})} - \frac{q}{\lambda} \sum_{j \in N_1(K)} (|u_j| - 1)$ and $(b(K), p(K), \lambda_K) \triangleq \arg\max_{b,p,\lambda} G(K, \lambda, b, p)$. In view of (2.37), one can verify that $b(K) = \frac{|N_0| + |N_1(K)|}{n}$ and $p(K) = \frac{|N_0|}{|N_0| + |N_1(K)|}$, and whenever $K > 1$,

$$\lambda_K = \arg\max_{0 \leq \lambda \leq \infty} L(K, \lambda). \tag{2.39}$$

When $K = 1$, $G(K, \lambda, b, p)$ does not depend on λ and hence λ_1 can be selected arbitrarily.

We are now led to determining λ_K for each $1 < K \leq A$. At this point, we invoke the following lemma.

Lemma 1 *Let*

$$g(t) \triangleq \frac{e^{-t}}{1-e^{-t}} - \frac{Ke^{-Kt}}{1-e^{-Kt}}.$$

Then for any $1 < K \leq A$, $L(K, \frac{q}{t})$ *as a function of* $t > 0$ *is strictly concave, and for any* $K > 1$, $g(t)$ *is strictly decreasing over* $t \in (0, \infty)$, *and* $\lim_{t\to 0^+} g(t) = \frac{K-1}{2}$ *and* $\lim_{t\to\infty} g(t) = 0$.

We prove Lemma 1 as follows. First note that $g(t)$ can be rewritten as $g(t) = K - 1 + \frac{1}{1-e^{-t}} - \frac{K}{1-e^{-Kt}}$. Its derivative is equal to

$$\begin{aligned} g'(t) &= \frac{-e^{-t}}{(1-e^{-t})^2} + \frac{K^2 e^{-Kt}}{(1-e^{-Kt})^2} \\ &= -\frac{e^{-t}}{(1-e^{-Kt})^2}\left[(\sum_{i=0}^{K-1} e^{-it}) - Ke^{-(K-1)t/2}\right] \\ &\quad \cdot \left[(\sum_{i=0}^{K-1} e^{-it}) + Ke^{-(K-1)t/2}\right]. \end{aligned} \tag{2.40}$$

It is not hard to verify that

$$\left[(\sum_{i=0}^{K-1} e^{-it}) - Ke^{-(K-1)t/2}\right] = \sum_{i=0}^{K_{\rm L}} (e^{-it/2} - e^{-\frac{K-1-i}{2}t})^2 > 0$$

whenever $K > 1$, where $K_{\rm L} = \text{floor}(\frac{K}{2}) - 1$. This, together with (2.40), implies that $g'(t) < 0$ for any $t > 0$ whenever $K > 1$. Hence $g(t)$ is strictly decreasing over $t \in (0, \infty)$.

Next we have

$$\begin{aligned} \lim_{t\to 0^+} g(t) &= \lim_{t\to 0^+} \frac{e^{-t}}{1-e^{-Kt}}\left[\sum_{i=0}^{K-1} e^{-it} - K \cdot e^{-(K-1)t}\right] \\ &= \frac{1}{2}(K-1). \end{aligned}$$

Finally, the strict concavity of $L(K, \frac{q}{t})$ as a function of t follows from (2.40) and the fact that

$$\frac{\partial^2 L(K, \frac{q}{t})}{\partial t^2} = |N_1(K)| g'(t).$$

This completes the proof of Lemma 1.

Computing the derivative of $L(K,\lambda)$ with respect to λ and setting it to 0 yields

$$\frac{e^{-q/\lambda}}{1-e^{-q/\lambda}} - K\frac{e^{-Kq/\lambda}}{1-e^{-Kq/\lambda}} - C = 0 \tag{2.41}$$

where $C = \frac{1}{|N_1(K)|}\sum_{j\in N_1(K)}(|u_j|-1)$. In view of Lemma 1, then it follows that (1) when $C=0$, $\lambda_K = 0$; (2) when $C \geq \frac{K-1}{2}$, $\lambda_K = \infty$; and (3) when $0 < C < \frac{K-1}{2}$, λ_K is the unique solution to (2.41). In Case (3), the iterative procedure described below in Algorithm 3 can be used to find the unique root of (2.41).

Algorithm 3 Estimating λ_K for a truncated geometric model

1: Select $\lambda_1 > 0$ arbitrarily if $K = 1$.

2: Compute $C = \frac{1}{|N_1(K)|}\sum_{j\in N_1(K)}(|u_j|-1)$.

3: Set $\lambda_K = 0$ if $C = 0$; or Set $\lambda_K = \infty$ if $C \geq \frac{K-1}{2}$.

4: Otherwise, set $C_0 = C$ and $\lambda^{(0)} = q/\ln\frac{1+C_0}{C_0}$.

5: For $i \geq 1$, compute

$$\begin{cases} C_i = & C + \frac{K}{e^{Kq/\lambda^{(i-1)}}-1} \\ \lambda^{(i)} = & \frac{q}{\ln\frac{1+C_i}{C_i}} \end{cases} \tag{2.42}$$

6: Repeat Step 5 until $\lambda^{(i)} - \lambda^{(i-1)} < \epsilon$, where $\epsilon > 0$ is a small prescribed threshold.

With λ_K computed by Algorithm 3, the optimal K^* will be obtained by solving

$$K^* = \arg\max_{1\leq K\leq A} G(K, b(K), p(K), \lambda_K). \tag{2.43}$$

Accordingly, we have $b^* = b(K^*)$, $p^* = p(K^*)$, and $\lambda^* = \lambda_{K^*}$.

Convergence and Complexity Analysis

In parallel with Algorithm 2, Algorithm 3 also converges exponentially, as summarized in Theorem 3.

Theorem 3 *Assume that* $0 < C < (K-1)/2$. *Then* $\lambda^{(i)}$ *computed in Step 5 of Algorithm 3 strictly increases and converges exponentially fast to* λ_K *as* $i \to \infty$.

We then provide the proof of Theorem 3. Arguments similar to those in the proof of Theorem 2 can be used to show that $\lambda^{(i)}$ is upper bounded by λ_K, strictly increases, and converges to λ_K as $i \to \infty$. Therefore what remains is to show that the convergence is exponentially fast. To this end, let $h(\lambda) \triangleq \frac{e^{-q/\lambda}}{1-e^{-q/\lambda}}$. In view of (2.42), it follows that

$$\begin{aligned} h(\lambda^{(i+1)}) &= C + \frac{K}{e^{Kq/\lambda^{(i)}} - 1} \\ &= C + Kh(\lambda^{(i)}/K) \end{aligned}$$

and hence

$$\begin{aligned} & h(\lambda^{(i+1)}) - h(\lambda^{(i)}) \\ &= Kh(\lambda^{(i)}/K) - Kh(\lambda^{(i-1)}/K) \\ &= \frac{Kh(\lambda^{(i)}/K) - Kh(\lambda^{(i-1)}/K)}{h(\lambda^{(i)}) - h(\lambda^{(i-1)})} \left[h(\lambda^{(i)}) - h(\lambda^{(i-1)})\right] \\ &\leq \delta \left[h(\lambda^{(i)}) - h(\lambda^{(i-1)})\right] \end{aligned}$$

where $\delta = \sup\{\frac{Kh(\lambda/K)-Kh(\nu/K)}{h(\lambda)-h(\nu)} : \lambda, \nu \in [\lambda^{(0)}, \lambda_K], \lambda \neq \nu\}$. In view of Lemma 1 and its proof (particularly (2.40)), it is not hard to verify that $0 < \delta < 1$. Therefore, as $i \to \infty$, $h(\lambda^{(i)})$ converges to $h(\lambda_K)$ exponentially fast. Since the derivative of $h(\lambda)$ is positive over $\lambda \in [\lambda^{(0)}, \lambda_K]$ and bounded away from 0, it follows that $\lambda^{(i)}$ also converges to λ_K exponentially fast. This competes the proof of Theorem 3.

The complexity of computing the ML estimate of the GMTCM parameters comes from two parts. The first part is to evaluate the cost of (2.37) over a set of K. The second part is to compute λ_K for every K using the Algorithm 3. Note that C in Algorithm 3 can be easily pre-computed for interesting values of K. Thus, the main complexity of

Algorithm 3 is to evaluate the two simple equations in (2.42) for a small number of times in light of the exponential convergence, which is generally negligible. Essentially, the major complexity for the parameter estimation by Algorithms 3 is to collect the data histogram $\{h_j, j = 1, \cdots, A\}$ once. Compared with the complexity of GG parameters estimation in [33] as shown in (2.3), where the data samples and the parameters to be estimated are closely tied together as in the $\sum_{i=1}^{n} |Y_i|^{\beta} \log |Y_i|$ term and the $\beta \sum_{i=1}^{n} |Y_i|^{\beta}$ term, the complexity of parameter estimation in the case of GMTCM is significantly lower.

2.5 Experimental results on Tests of modelling Accuracy

This section presents experimental results obtained from applying TCMs to both continuous and discrete DCT coefficients and compare them with those from the Laplacian and GG models. In general, Laplacian is very simple and easy to apply, yet has an inferior modelling accurancy; GGD is very complicated, but provides a superior modelling accuracy.

2.5.1 Test conditions and test materials

Two criteria are applied in this paper to test modelling accuracy: the χ^2 test, as defined in (2.8), and the KL divergence by (2.7). When comparison is conducted, a factor w_d is calculated to be the percentage of DCT frequencies among all tested AC positions that are in favor of one model over another model in terms of having a smaller KL divergence from the data distribution. Another factor w_{χ^2} is defined in a similar way, except that the comparison is carried out based on the χ^2 test results for individual frequencies.

Three sets of testing images are deliberately selected to cover a variety of image content. The first set includes 9 512×512 images used in JPEG standardization with faces, animals, buildings, landscapes, etc, referred to 'Bird', 'Boat', 'Fish', 'couple/Cp', 'Hill', 'Lena', 'baboon/Bb', 'mountain/Bt', and 'pepper/Pp', respectively. The second set has five 1080p high-definition frames selected from the first frame of each class-B sequences used for HEVC standardization tests [32], named as 'BQTerrace', 'BasketballDrive', 'Cactus', 'Kimono', and 'Parkview', and referred to as 'B1', 'B2', 'B3', 'B4', and 'B5', respectively, hereafter. The third set is taken from the first frame of four class-F sequences used for HEVC screen content tests, named as 'SlideEditing', 'SlideShow', 'ChinaSpeed', and 'BasketballText', and referred to as 'SE', 'SS', 'CS', and 'BbT', respectively, hereafter.

Tests for continuous DCT coefficients were conducted by computing 8×8 DCT using floating point operations. In our tests for discrete DCT coefficients, a raw image was first compressed using a Matlab JPEG codec with various quality factors (QF) ranging from 100, 90, 80, to 70; the resulting quantized DCT coefficients and corresponding quantization step sizes were then read from obtained JPEG files.

Tests were carried out for six different models: the Cauchy model, the Laplacian model, GG model, GGTCM, LPTCM, and GMTCM. GGTCM was applied only to continuous DCT coefficients. On the other hand, GMTCM is applicable only to discrete coefficients. The Laplacian and GG models were applied to both continuous and discrete DCT coefficients.

2.5.2 Overall comparisons for each image

Table 2.1 shows comparisons between the Cauchy model and GGD based on both the KL test and the χ^2 test. When KL test is used, only 5% on average of all 63 AC frequencies

Table 2.1: Comparing Cauchy model with GGD (continuous DCT).

	Bird	Boat	Fish	Cp	Hill	Lena	Bb	Mt	Pp
w_d (%)	0	0	5	0	0	16	0	0	21
w_{χ^2} (%)	78	57	6	63	32	67	14	29	83

are in favor of the Cauchy model. Although the result by χ^2 test shows some merits of the Cauchy model for fitting the tail part, the inferior result by KL test shows that the Cauchy model does not model the main portion well. To some extend, the discrepancy between the KL test results and the χ^2 test results indicates that the Cauchy model retains a flat tail at a cost of losing accuracy for the main portion. Furthermore, it is observed that the Cauchy model in general offers a modelling accuracy comparable to GGD when the data fits the GGD model with a shape parameter β within a range of [0.45, 0.55]. Nevertheless, for the 9 images in the test set 1, β varies in a range of [0.3, 1.3]. (Note that Laplacian is a special case of GGD with $\beta = 1$.)

In the continuous case, the GGTCM outperforms the GG model, the LPTCM outperforms the Laplacian model, and the GG models outperforms the Laplacian model in general, as one would expect. An interesting comparison in this case is between the GG model and LPTCM. Table 2.2 shows the percentage w_{χ^2} of frequencies among 63 AC positions that are in favor of the LPTCM over the GG model for each of 9 images in Set 1 in terms of the χ^2 metric. For example, for the image 'Bird', in terms of the χ^2 metric, the LPTCM is better than the GG model for 60 out of 63 frequencies; for the image 'Lena', the LPTCM is better than the GG model for 36 out of 63 frequencies. Overall, it would be fair to state that the LPTCM and GG model behave similarly in terms of modelling accuracy. And yet, the LPTCM has much lower computation complexity than the GG model.

In the discrete case, comparisons were conducted among the GMTCM, GG model, and

Table 2.2: Comparing LPTCM with GGD (continuous DCT).

	Bird	Boat	Fish	Cp	Hill	Lena	Bb	Mt	Pp
w_d (%)	90	25	21	25	37	41	22	38	54
w_{χ^2} (%)	95	49	8	52	51	57	40	79	81

Table 2.3: Overall comparisons between the GMTCM and GG model for all images coded using JPEG with QF= 100.

	Bird	Boat	Fish	Cp	Hill	Lena	Bb	Mt	Pp	SE	SS	CS	BbT	B1	B2	B3	B4	B5
w_d (%)	95	38	100	44	59	60	49	48	71	95	78	8	48	52	83	62	40	60
w_{χ^2} (%)	98	57	100	59	67	67	52	83	84	83	89	8	62	52	89	71	65	65

Laplacian model in terms of both the divergence distance and χ^2 value. As expected, the GMTCM is always better than the Laplacian model according to both the divergence distance and χ^2 value, and hence the corresponding results are not included here. For the comparison between the GMTCM and GG model, results are shown in Tables 2.3, 2.4, 2.5, and 2.6 for quantized DCT coefficients from JPEG coded images with various QFs. In Tables 2.3, all 63 AC positions were tested; in Tables 2.4, 2.5, and 2.6, all AC positions with 6 or more different non-zero AC coefficient magnitudes were tested. These tables show that when all quantization step sizes are 1, corresponding to QF= 100, the comparison between the GMTCM and GG model is similar to that between the LPTCM and GG model, i.e., their performances are close to each other. However, with quantization step sizes increasing, the GMTCM starts to outperform the GG model significantly, as shown in Tables 2.4, 2.5, and 2.6, for all tested images.

Table 2.4: Overall comparisons between the GMTCM and GG model for all images coded using JPEG with QF= 90.

	Bird	Boat	Fish	Cp	Hill	Lena	Bb	Mt	Pp	SE	SS	CS	BbT	B1	B2	B3	B4	B5
w_d (%)	89	73	97	69	82	75	79	83	85	100	92	79	86	78	93	76	85	75
w_{χ^2} (%)	95	73	98	67	80	71	79	84	87	98	90	78	79	73	91	73	89	75

Table 2.5: Overall comparisons between the GMTCM and GG model for all images coded using JPEG with QF= 80.

	Bird	Boat	Fish	Cp	Hill	Lena	Bb	Mt	Pp	SE	SS	CS	BbT	B1	B2	B3	B4	B5
w_d (%)	98	76	98	79	88	79	86	81	86	100	97	87	86	70	94	80	91	85
w_{χ^2} (%)	98	78	90	79	82	74	84	83	86	100	95	81	86	68	91	78	95	79

Table 2.6: Overall comparisons between the GMTCM and GG model for all images coded using JPEG with QF= 70.

	Bird	Boat	Fish	Cp	Hill	Lena	Bb	Mt	Pp	SE	SS	CS	BbT	B1	B2	B3	B4	B5
w_d (%)	94	83	95	82	79	77	90	98	84	98	98	87	89	73	97	85	95	87
w_{χ^2} (%)	97	83	89	79	91	80	84	94	87	98	97	87	89	75	94	85	95	83

2.5.3 Comparisons of χ^2 among three models for individual frequencies

In the above overall comparisons, Table 2.3 shows that the GMTCM and GG model are close, while the GMTCM wins the majority over the GG model for all other cases as shown in Tables 2.4-2.6. We now zoom in to look at the χ^2 values for all tested frequency positions for several representative images: (1) 'Bird' which is strongly in favor of the GMTCM in Table 2.3; (2) 'CS' which is strongly in favor of the GG model in Table 2.3; and (3) 'Boat' for which the GMTCM and GG model tie more or less in Table 2.3. The respective χ^2 scores are presented in Figures 2.4, 2.6, and 2.7, respectively.

From Figures 2.4, 2.6, and 2.7, it is fair to say that (1) the GMTCM dramatically improves the modelling accuracy over the Laplacian model; (2) when the GMTCM is better than the GG model, χ^2_{GMTCM} is often much smaller, up to 15658 times smaller, than χ^2_{GGD}; and (3) when the GG model is better than the GMTCM, the difference between χ^2_{GMTCM} and χ^2_{GGD} is not as significant as one would see in Case (2)—for example, in Figure 2.7, χ^2_{GGD} is only less than 10 times smaller than χ^2_{GMTCM}.

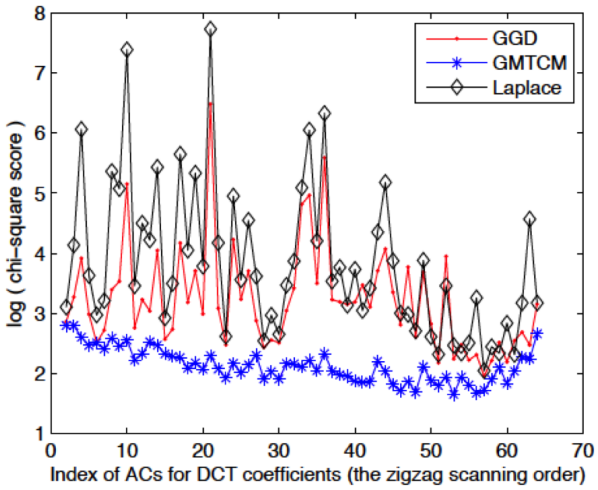

Figure 2.4: The χ^2 scores by GGD, GMTCM, and Laplacian model for AC coefficients from JPEG-coded image Bird with QF= 100.

Note that Figure 2.5 shows the KL convergence for the 'Bird' image. It demonstrates that the results by the chi-square test are consistent with that by the KL convergence, as supported by the consistence between w_{χ^2} and w_d in the overall comparisons in Tables 2.3 to 2.6.

Another interesting result is observed in Figure 2.8, which shows the χ^2 values for JPEG coded 'CS' image with QF=90. Compared with the case where the source is JPEG coded with higher fidelity QF=100 as shown in Figure 2.7, most AC coefficients now show better modelling accuracy by the GMTCM than by the GG model.

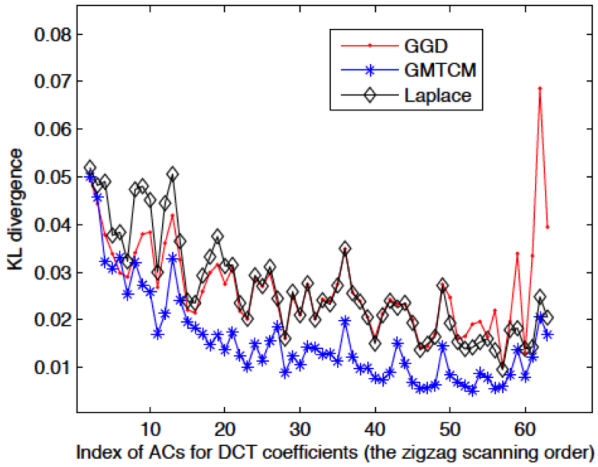

Figure 2.5: The KL divergence by GGD, GMTCM, and Laplacian model for AC coefficients from JPEG-coded image Bird with QF= 100.

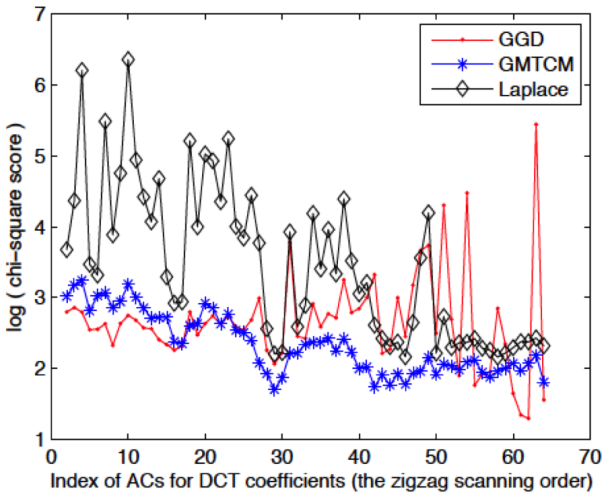

Figure 2.6: The χ^2 scores by GGD, GMTCM, and Laplacian model for AC coefficients from JPEG-coded image Boat with QF= 100.

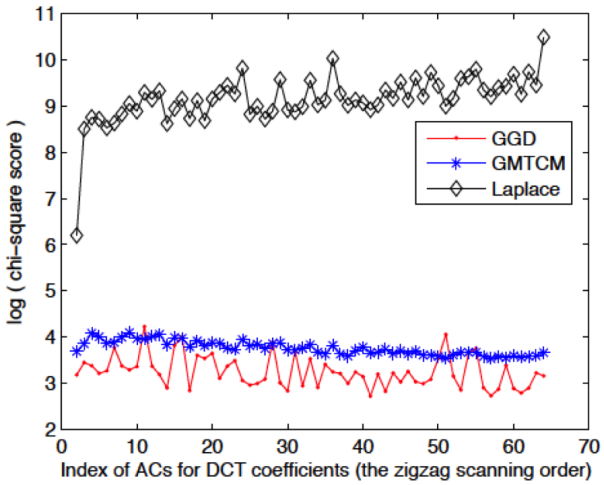

Figure 2.7: The χ^2 scores by GGD, GMTCM, and Laplacian model for AC coefficients from JPEG-coded image CS with QF= 100.

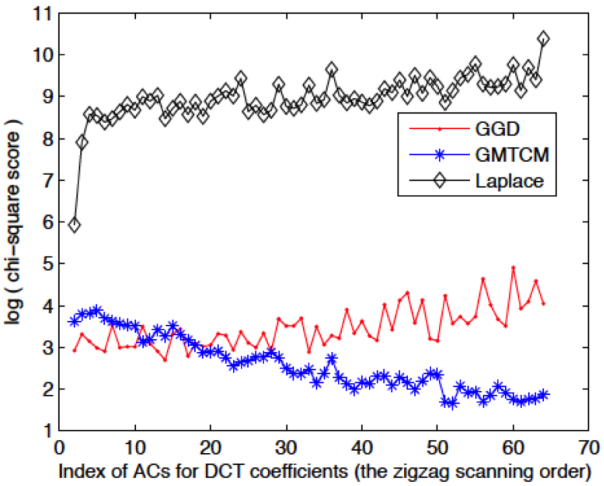

Figure 2.8: The χ^2 scores by GGD, GMTCM, and Laplacian model for AC coefficients from JPEG-coded image CS with QF= 90.

2.6 Data reduction capability of Transparent Composite Model

The segmentation between a parametric distribution for the main portion and the uniform distribution for the heavy tail portion naturally intrigues us to understand their respective physical meanings in image domain. Inliers in the main portion contains most of the information[1], which should undoubtedly represent a blurred version of the original image (since some information is missing). But how about outliers in the heavy tail portion? What kind of information it conveys in an image?

To investigate this, we force all the inliers (as well as DCs) to be zeros, i.e., only keep outliers, and then perform inverse DCT, yielding an image that we call an outlier image. Two outlier images obtained by using the LPTCM for 512×512 Lena and Airplane (F-16) are shown in the right panels of Fig.2.9 and Fig.2.10, along with their original images shown in their respective left panels (more examples of outlier images can be found in [76] and [75]). As demonstrated in the outlier images, DCT coefficients in the heavy tail identified by a TCM are truly outliers, and the outlier image reveals some unique global features of the original image, such as edges. That is to say, although they are usually statistically insignificant—on average around 1.2% of the total AC coefficients of the image as indicated by our experiments, outliers (or outlier image) are of perceptually importance. Compared with the original image, the outlier image achieves dramatic dimension reduction, while still capturing the main skeleton of the former. This feature of outliers in the LPTCM can be used to design DCT-based lossy compression algorithms. As we show later in Chapter 4, outliers in the LPTCM are taken good advantage of in designing conditional quantization and conditional entropy coding. In addition, with the non-linear data reduction capability,

[1]In most of the cases, more than 98% of the DCT coefficients are identified as inliers by a TCM.

a TCM may also be applied in image analysis/understanding, etc., as suggested in [76] and [75].

Figure 2.9: Original image (Left) and outlier image (Right) of 512×512 Lena

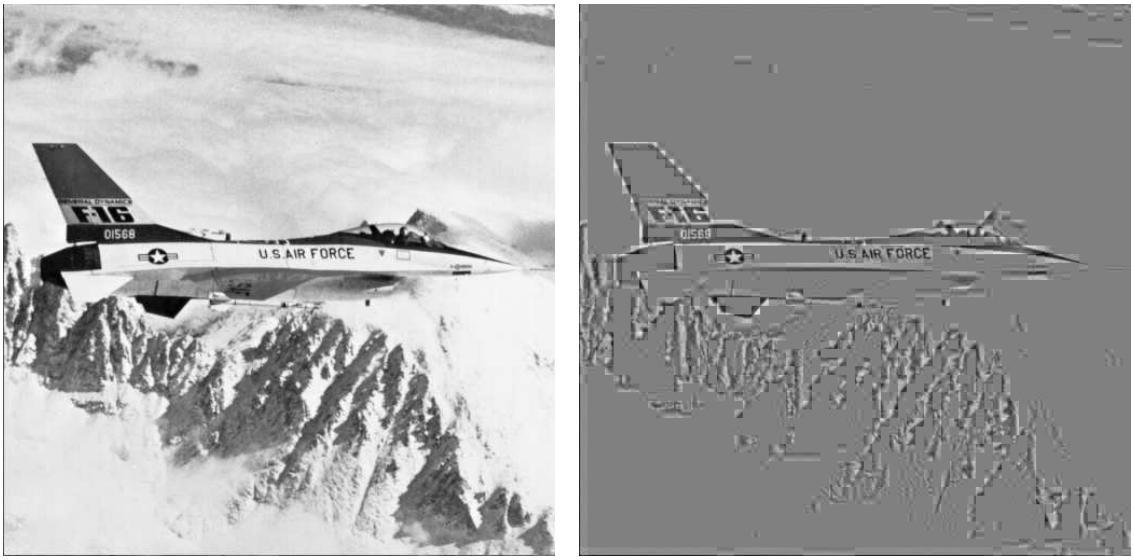

Figure 2.10: Original image (Left) and outlier image (Right) of 512×512 Airplane (F-16)

2.7 Chapter Summary

To better capture the heavy tail phenomenon in DCT coefficients and its perceptual importance, in this chapter we have proposed a new model dubbed TCM for modelling DCT coefficients, which separates the tail portion of DCT coefficients from the main portion of DCT coefficients and uses a different distribution to model each portion: a uniform distribution for the tail portion and a parametric distribution such as truncated Laplacian and geometric distributions for the mail portion. Efficient online algorithms with global convergence have been proposed to compute the ML estimates of the parameters in the TCM. It has been shown that for real-valued continuous AC coefficients, the LPTCM matches up to pure GG models in term of modelling accuracy, but with simplicity and practicality similar to those of pure Laplacian models. On the other hand, for discrete/integer DCT coefficients, the GMTCM models AC coefficients more accurately than pure Laplacian models and GG mdoels in majority cases while having simplicity and practicality similar to those of pure Laplacian models. In addition, it has been demonstrated that the tail portion identified by the LPTCM gives rise to an image called an outlier image, which, on one hand, achieves dramatic dimension reduction in comparison with the original image, and on the other hand preserves perceptually important unique global features of the original image. The LPTCM will be applied to design DCT-based lossy image compression algorithms in Chapter 4.

Chapter 3

Quantization Table Design Revisited for Image/Video compression

This chapter revisits the quantization table design problem in soft-decision quantization (SDQ) settings. Section 3.1 reviews some related quantization table design methods proposed in traditional hard-decision quantization (HDQ) settings. Section 3.2 formulates our new quantization table design problem, which is solve in Section 3.3 as our theoretical contribution. A statistical-model-based algorithm is discussed in Section 3.4, and experimental results are given in Section 3.5. Finally, we summarize this chapter in Section 3.6.

3.1 Literature review

In DCT-based image/video compression, a quantization table is usually used to provide different quantization step sizes to different frequencies. After each block is DCT trans-

formed, each of the resulting transform coefficients is quantized uniformly according to a quantization step size; all quantization step sizes used at different transform frequencies together form a quantization table. Quantized coefficients are then scanned into a 1-D sequence and finally encoded losslessly.

Traditionally, as long as a quantization table is fixed, the corresponding quantization process is determined, where the quantized value of a transform coefficient depends only on its quantization step size and the transform coefficient itself. Such a quantization process is now referred to as HDQ. In this case, once the subsequent lossless coding method is given, quantization table design is equivalent to quantizer design.

In the past two decades, quantization table design has been well studied in HDQ settings [19], [62], [42]. It is generally formulated as the following optimization problem[1]

$$\inf_{Q} R(Q), \text{ s.t. } D(Q) \leq D_T, \tag{3.1}$$

where Q represents a quantization table, R and D are the resulting rate and distortion respectively, and D_T denotes the target distortion. With the lossless coding method fixed, both R and D are functions of Q only. However, since $R(Q)$ is generally a very complicated function of Q, finding an optimal solution or even a good approximate solution to (3.1) is computationally expensive. On the other hand, if $R(Q)$ is inaccurately approximated, the solution to the resulting modified optimization problem would be far away from the solution to (3.1). For example, for JPEG encoding [1], Huang and Meng [19] proposed a quantization table optimization method where DCT coefficients are modelled by a Laplacian distribution. The rate (as well as distortion) is estimated by close-form formulas based on the statistic model, where the default Huffman code length specified by JPEG is assumed. The performance gain offered by their optimizer is limited mainly because of the

[1]The optimization problem can also be formulated as minimizing the distortion at a given rate.

inaccuracy in approximating the rate. To solve this problem, Wu and Gersho [62] proposed to evaluate the rate using the real coding rate with a greedy, steepest-descent algorithm. It achieves better RD performance but at the expense of extremely high computational complexity, since the actual encoding is performed in each iteration of the algorithm. To avoid going through the actual encoding repeatedly, Ratnakar and Livny [42] later on developed a comparatively efficient [2] JPEG quantization table optimizer using a trellis-based method where the rate is estimated by the empirical entropy of quantized DCT coefficients, rather than the coding rate of run-size pairs as in [62]. Their scheme achieves RD performance similar to what was reported in [62], which represents the best JPEG coding performance so far when HDQ is assumed.

Recently, a more advanced quantization technique called SDQ has been developed [79], [73], [74], [72], [70], [71], [16], [64], [22]. Unlike the case of HDQ, the quantization process in SDQ is now tightly coupled with, and to some extent controlled by, the subsequent lossless coding method so that a better RD trade-off is achieved. Even with the same quantization table, the quantization process would be dramatically different if a different lossless coding method were used. Because of its superiority over HDQ, SDQ or its suboptimal version called rate distortion optimized quantization (RDOQ) [22] has been well adopted in both video coding standards H.264/AVC [31] and HEVC [32].

Given a lossless coding method, quantizer design under SDQ is not equivalent to quantization table design any more. Indeed, the quantizer design problem in SDQ can be separated into two sub-problems, i.e., quantization table design and SDQ design (see Section 3.2 for details). In the past, several researchers [42], [72], [14] studied these two design problems for JPEG encoding to some extent. In [14], Crouse and Ramchandran applied

[2] Compared with the algorithm in [62] it is more efficient, which is nevertheless considered to be computationally expensive as shown in Section 3.5 in this thesis.

the method proposed in [62] to design the quantization table followed by the algorithm proposed in [39] called optimal thresholding to optimize quantized DCT coefficients, which is actually a sub-optimal SDQ design for JPEG coding. In [42], Ratnakar and Livny employed their quantization table optimization method to initialize the thresholding algorithm in [39]. Optimal SDQ design problem for JPEG coding was later on solved by a graph-based algorithm in [72] by Yang and Wang. To address the optimal design of both the quantization table and SDQ, they further proposed an iterative algorithm, which achieves the best JPEG coding performance in the literature when SDQ is considered. However, unlike the graph-based algorithm, which is optimal, the iterative algorithm does not seem to have a global convergence, often leading to local minima; as such, its performance highly depends on the initial quantization table [3]. Indeed, Yang and Wang [72] used the quantization table optimizer in [42] to give an initial quantization table for their iterative algorithm in their best quantization scheme. Therefore, to a large extent, the quantization table design in SDQ has never been fully addressed.

3.2 Quantization table design–Problem formulation

In light of the increasing importance of SDQ in image and video coding, we are motivated to revisit quantization table design in the context of SDQ [68], [69]. In this section, we first formulate the new quantizer design problem for image/video coding where the quantization table optimization is separated as a sub-problem. We then show some advantages of the new formulation for the quantization table design problem over (3.1).

Suppose that an $N \times M$ transform is used, and there are in total n non-overlapping blocks of size $N \times M$ in each image or video frame. (Hereafter, each block of size $N \times M$

[3] This is true as well for the iterative algorithm called joint thresholding proposed in [14].

will be referred to as an $N \times M$-block.) Define a quantization table $Q = \{q_1, q_2, \cdots, q_L\}$, where q_k is the quantization step size for transform coefficients at the ith frequency position in a predefined scanning order, $i \in [1, L]$, where $L = N \times M$. Given Q, each transform coefficient at the ith frequency position would be reconstructed as jq_k for some $j \in \{0, \pm 1, \pm 2, \cdots, \pm\lfloor 1 + \frac{A}{q_k} \rfloor\}$, where A represents the maximum possible magnitude that transform coefficients could have. For example, $A = 1024$ in 8-bit JPEG encoding [1]. In this sense, we say that Q solely determines the reconstruction space, and would simply identify it with a reconstruction space in this thesis when there is no ambiguity. When SDQ is adopted, we can divide a quantization process into two parts: a quantization table or reconstruction space Q and a mapping function or quantizers Q'_Q that maps each sequence of transform coefficients at the ith frequency position, $i \in [1, L]$, into an index sequence of length n from $\{0, \pm 1, \pm 2, \cdots, \pm\lfloor 1 + \frac{A}{q_k} \rfloor\}^n$. Given a lossless coding method ϕ for index sequences, quantizer design under SDQ is equivalent to the following optimization problem:

$$\inf_{Q} \inf_{Q'_Q} R_\phi(Q, Q'_Q), \text{ s.t. } D(Q, Q'_Q) \leq D_T \tag{3.2}$$

where $R_\phi(Q, Q'_Q)$ denotes the number of bits per $N \times M$-block resulting from using ϕ to encode the index sequences given by Q'_Q, and $D(Q, Q'_Q)$ is the distortion per $N \times M$-block resulting from the SDQ process Q'_Q in conjunction with Q. Here and throughout the rest of the thesis, we assume the mean squared error distortion. In comparison of (3.2) with (3.1), there is a striking difference. In (3.1), once Q is given, the quantized value of each transform coefficient is determined, and so is $D(Q)$. In addition, assuming the lossless coding method ϕ is fixed, $R(Q)$ is a function of Q only. On the contrary, in (3.2), both the rate and distortion are functions of both Q and Q'_Q. Even when Q is given, the quantized value of each transform coefficient is still undecided, until the solution or an approximate solution to the inner minimization problem of (3.2) is found. Therefore, to a large extent,

the quantization process is controlled by the subsequent lossless coding method ϕ.

Convert (3.2) into the following unconstrained optimization problem

$$\inf_{Q} \inf_{Q'_Q} \left[D(Q, Q'_Q) + \theta R_\phi(Q, Q'_Q) \right] \tag{3.3}$$

where θ is the Lagrange multiplier denoting the relative trade-off between rate and distortion. Given Q, the solution to the inner minimization problem in (3.3) is generally referred to as SDQ. In principle, with an initial quantization table Q, (3.3) can be solved by an iterative algorithm (as in [72] in the case of JPEG encoding): (Step 1) fix Q and seek a SDQ solution Q'_Q to the inner minimization problem in (3.3); (Step 2) fix the resulting Q'_Q and seek a solution Q to the outer minimization problem in (3.3); and (Step 3) repeat Steps 1 and 2 until a convergence occurs. Although each of Steps 1 and 2 could be optimal itself—please refer to [73], [74], [72], [71], [16],[22] for the applications of these steps to image and video coding standards proposed so far—the iterative algorithm does not converge, in general, to a global optimum. As such, its performance highly depends on the initial Q.

In this thesis, we aim to determine the optimal Q in (3.3) or its approximation without determining explicitly the optimal Q'_Q in (3.3). In view of (3.3), optimal Q and Q'_Q are clearly related to each other and both depend on the lossless coding method ϕ. To overcome this difficulty, we shall consider ϕ that is universal and optimal, in the sense that the inner minimization in (3.2) can be approximated by the Shannon rate distortion functions of transform coefficients with respect to the alphabet $\{0, \pm 1q_k, \pm 2q_k, \cdots, \pm\lfloor 1 + \frac{A}{q_k} \rfloor q_k\}$ for all Q when n is large enough. From universal lossy source coding theory [79], [81], [80], such lossless coding methods exist. The advantage of this approach is two fold. First, it makes the problem (3.2) or equivalently (3.3) tractable. Second, it makes our solution Q to be independent of any specific lossless coding method, which often varies from one application

to another, which in turn makes our solution Q widely applicable to many practical image and video coding problems where quantization tables are used, as a good initial Q.

3.3 Quantization table design–Problem solution

Following the approach alluded to at the end of Section 3.2, we now derive the optimal Q in (3.2) or its approximation without determining explicitly the optimal Q'_Q. As aforementioned, we will assume that ϕ is universal and optimal so that the inner minimization in (3.2) can be approximated by the Shannon rate distortion functions of transform coefficients with respect to the alphabet $\{0, \pm 1q_k, \pm 2q_k, \cdots, \pm\lfloor 1 + \frac{A}{q_k}\rfloor q_k\}$ for all Q when n is large enough.

Rewrite (3.2) as

$$\inf_{Q} \inf_{Q'_Q : D(Q,Q'_Q) \le D_T} R_\phi(Q, Q'_Q). \tag{3.4}$$

We model transform coefficients across different frequencies as independent random sources $\{X_k\}_{k=1}^L$ with certain distributions, where each X_k, $1 \le k \le L$, is a sequence $X_k = \{X_k(i)\}_{i=1}^n$ of length n, representing all transform coefficients at the ith frequency position, and further regard $D(Q, Q'_Q)$ and $R_\phi(Q, Q'_Q)$ in (3.4) as the average distortion and rate (in bits) of the L sources $\{X_k\}_{k=1}^L$, respectively. Under this assumption, transform coefficients can be optimally quantized and encoded by separately quantizing and encoding each source X_k, $1 \le k \le L$. Without loss of generality, we further assume that each source X_k has a zero mean; otherwise, the mean could be subtracted first. As such, the inner minimization

in (3.4) can be rewritten as

$$\begin{aligned}
&\inf_{Q'_Q: D(Q,Q'_Q)\le D_T} R_\phi(Q, Q'_Q) \\
&= \inf_{\substack{\{D_k\}_{k=1}^L \\ \sum_{k=1}^L D_k = D_T}} \inf_{\substack{Q'_{q_k}: D(X_k,q_k,Q'_{q_k})\le D_k \\ 1\le k\le L}} \sum_{k=1}^{L} R_\phi(X_k, q_k, Q'_{q_k}) \\
&= \inf_{\substack{\{D_k\}_{k=1}^L \\ \sum_{k=1}^L D_k = D_T}} \sum_{k=1}^{L} \left[\inf_{\substack{Q'_{q_k} \\ D(X_k,q_k,Q'_{q_k})\le D_k}} R_\phi(X_k, q_k, Q'_{q_k}) \right] \\
&= \inf_{\substack{\{D_k\}_{k=1}^L \\ \sum_{k=1}^L D_k = D_T \\ D_k \ge D(X_k,q_k)}} \sum_{k=1}^{L} \left[\inf_{\substack{Q'_{q_k} \\ D(X_k,q_k,Q'_{q_k})\le D_k}} R_\phi(X_k, q_k, Q'_{q_k}) \right] \qquad (3.5)
\end{aligned}$$

where Q'_{q_k}, $1 \le k \le L$, is a mapping from the set of sequences of n transform coefficients to the set $\{0, \pm1, \pm2, \cdots, \pm\lfloor 1 + \frac{A}{q_k} \rfloor\}^n$ and represents SDQ for the source X_k, $D(X_k, q_k, Q'_{q_k})$ denotes the average distortion per transform coefficient between X_k and the reconstruction sequence given by Q'_{q_k} and q_k (i.e., $Q'_{q_k} \times q_k$), $R_\phi(X_k, q_k, Q'_{q_k})$ denotes the corresponding rate in bits per transform coefficient for X_k, and $D(X_k, q_k)$ is the minimal average distortion per transform coefficient for the source X_k, which is achievable with the reconstruction space $\{0, \pm1q_k, \pm2q_k, \cdots, \pm\lfloor 1 + \frac{A}{q_k} \rfloor q_k\}^n$. Note that $D(X_k, q_k)$ is actually equal to the average distortion resulting from the HDQ of X_k with the quantization step size q_k. The last equality in (3.5) follows from the fact that when $D_k < D(X_k, q_k)$, the set of Q'_{q_k} with $D(X_k, q_k, Q'_{q_k}) \le D_k$ is empty and hence the corresponding inner minimization in (3.5) is ∞. At this point, we invoke universal redundancy results from lossy source coding theory [80], which say that when ϕ is universal and optimal,

$$\begin{aligned}
&\inf_{Q'_{q_k}: D(X_k,q_k,Q'_{q_k})\le D_k} R_\phi(X_k, q_k, Q'_{q_k}) \\
&= R_{X_k}^{q_k}(D_k) + c\left(1 + 2\lfloor 1 + \frac{A}{q_k} \rfloor\right) \frac{\ln n}{n} + o(\frac{\ln n}{n}) \qquad (3.6)
\end{aligned}$$

where $R_{X_k}^{q_k}(D_k)$ is the Shannon rate distortion function of X_k with respect to the reconstruction alphabet $\{0, \pm 1q_k, \pm 2q_k, \cdots, \pm\lfloor 1+\frac{A}{q_k}\rfloor q_k\}$, and $c = O(1)$ is a positive bounded term. Note that $1+2\lfloor 1+\frac{A}{q_k}\rfloor$ is simply the size of the alphabet $\{0, \pm 1q_k, \pm 2q_k, \cdots, \pm\lfloor 1+\frac{A}{q_k}\rfloor q_k\}$, which is proportional to $1/q_k$. By absorbing some positive bounded term into $c = O(1)$, we can rewrite (3.6) as

$$\begin{aligned} &\inf_{Q'_{q_k}: D(X_k, q_k, Q'_{q_k}) \le D_k} R_\phi(X_k, q_k, Q'_{q_k}) \\ &= R_{X_k}^{q_k}(D_k) + \frac{c}{q_k}\frac{\ln n}{n} + o(\frac{\ln n}{n}). \end{aligned} \tag{3.7}$$

Combining (3.7) and (3.5) with (3.4) yields

$$\begin{aligned} &\inf_{Q} \inf_{Q'_Q: D(Q, Q'_Q) \le D_T} R_\phi(Q, Q'_Q) \\ &= \inf_{Q} \inf_{\substack{\{D_k\}_{k=1}^L: \sum_{k=1}^L D_k = D_T \\ D_k \ge D(X_k, q_k)}} \sum_{k=1}^{L} \Big[R_{X_k}^{q_k}(D_k) \\ &\quad + \frac{c}{q_k}\frac{\ln n}{n} + o(\frac{\ln n}{n}) \Big]. \end{aligned} \tag{3.8}$$

Since there is no analytic formula for $R_{X_k}^{q_k}(D_k)$ in general, to continue with (3.8), we further lower bound $R_{X_k}^{q_k}(D_k)$ by the Shannon lower bound to the rate distortion function of X_k [12]:

$$\begin{aligned} &R_{X_k}^{q_k}(D_k) \\ &\ge R_{X_k}^{(SL)}(D_k) \\ &\triangleq \max\{H(X_k) - \frac{1}{2}\log 2\pi e D_k, 0\} \qquad (3.9) \\ &= \begin{cases} H(X_k) - \frac{1}{2}\log 2\pi e D_k & \text{if } \hat{\sigma}_k^2 > D_k \\ 0 & \text{otherwise} \end{cases} \qquad (3.10) \end{aligned}$$

where $H(X_k)$ is the differential entropy of X_k, and $\hat{\sigma}_k^2$ is chosen such that $H(X_k) = \frac{1}{2}\log 2\pi e\hat{\sigma}_k^2$. (Note that according to the maximum differential entropy lemma [12], $\hat{\sigma}_k^2 \leq \sigma_k^2$, where σ_k^2 is the variance of X_k.) Plugging (3.9) into (3.8) yields

$$\begin{aligned}
&\inf_{Q} \inf_{Q'_Q: D(Q,Q'_Q)\leq D_T} R_\phi(Q, Q'_Q) \\
&\geq \inf_{Q} \inf_{\substack{\{D_k\}_{k=1}^L: \sum_{k=1}^L D_k = D_T \\ D_k \geq D(X_k, q_k)}} \sum_{k=1}^{L} \Big[R_{X_k}^{(SL)}(D_k) \\
&\qquad + \frac{c}{q_k}\frac{\ln n}{n} + o(\frac{\ln n}{n}) \Big].
\end{aligned} \tag{3.11}$$

In (3.11), one can limit Q to those satisfying

$$\sum_{k=1}^{L} D(X_k, q_k) \leq D_T \tag{3.12}$$

since the distortion profile D_k satisfying $\sum_{k=1}^L D_k = D_T$ with $D_k \geq D(X_k, q_k)$, $1 \leq k \leq L$, does not exist if (3.12) is not valid.

We are now led to solve the following optimization problem instead

$$\inf_{Q} \inf_{\substack{\{D_k\}_{k=1}^L: \sum_{k=1}^L D_k = D_T \\ D_k \geq D(X_k, q_k)}} \sum_{k=1}^{L} \Big[R_{X_k}^{(SL)}(D_k) + \frac{c}{q_k}\frac{\ln n}{n} + o(\frac{\ln n}{n}) \Big]. \tag{3.13}$$

Given Q satisfying (3.12), solving the inner minimization in (3.13) is now equivalent to solving

$$\inf_{\substack{\{D_k\}_{k=1}^L: \sum_{k=1}^L D_k = D_T \\ D_k \geq D(X_k, q_k)}} \sum_{k=1}^{L} \Big[R_{X_k}^{(SL)}(D_k) \Big] \tag{3.14}$$

since the last two terms in (3.13) do not depend on D_k, $1 \leq k \leq L$. Note that $R_{X_k}^{(SL)}(D_k)$ given in (3.9) is a convex function of D_k. According to the Kuhn-Tucker conditions, the

minimization in (3.14) or equivalently the inner minimization in (3.13) is achieved when

$$D_k = D_k(Q) \triangleq \begin{cases} d & \text{if } D(X_k, q_k) \le d \le \hat{\sigma}_k^2 \\ D(X_k, q_k) & \text{if } d < D(X_k, q_k) \\ \hat{\sigma}_k^2 & \text{if } d > \hat{\sigma}_k^2 \end{cases} \tag{3.15}$$

where d is chosen so that

$$\sum_{k=1}^{L} D_k(Q) = D_T. \tag{3.16}$$

As D_T increases from $\sum_{k=1}^{L} D(X_k, q_k)$, the value of d satisfying (3.16) also increases until it hits the ceiling $\max\{\hat{\sigma}_k^2 : 1 \le k \le L\}$. The solution $D_k(Q)$ in (3.15) can be interpreted as gas pumping with caps from both top and bottom, as illustrated in Figure 3.1. Plugging (3.15) into (3.13), we have

$$\begin{aligned} & \inf_{Q} \inf_{\substack{\{D_k\}_{k=1}^{L}: \sum_{k=1}^{L} D_k = D_T \\ D_k \ge D(X_k, q_k)}} \sum_{k=1}^{L} \left[R_{X_k}^{(SL)}(D_k) + \frac{c}{q_k} \frac{\ln n}{n} + o(\frac{\ln n}{n}) \right] \\ = & \inf_{Q} \sum_{k=1}^{L} \left[R_{X_k}^{(SL)}(D_k(Q)) + \frac{c}{q_k} \frac{\ln n}{n} + o(\frac{\ln n}{n}) \right] \\ = & \inf_{Q} \left[\sum_{k=1}^{L} R_{X_k}^{(SL)}(D_k(Q)) + \sum_{k=1}^{L} \left(\frac{c}{q_k} \frac{\ln n}{n} + o(\frac{\ln n}{n}) \right) \right]. \end{aligned} \tag{3.17}$$

To continue, let us first investigate

$$\inf_{Q} \sum_{k=1}^{L} R_{X_k}^{(SL)}(D_k(Q)).$$

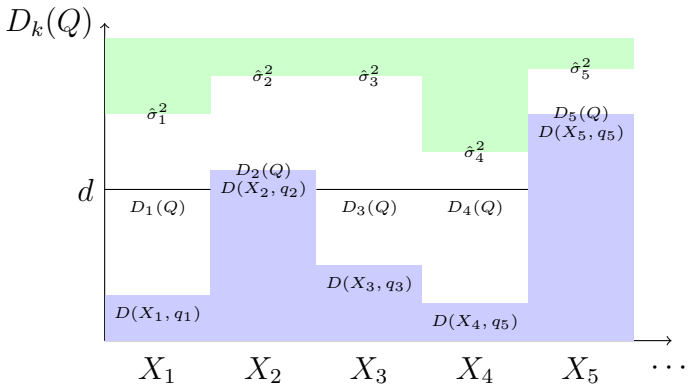

Figure 3.1: Gas pumping illustration for $D_k(Q)$.

In view of (3.14) to (3.16), it is not hard to see that

$$\begin{aligned}
&\inf_{Q} \sum_{k=1}^{L} R_{X_k}^{(SL)}(D_k(Q)) \\
&= \inf_{\{D_k\}_{k=1}^{L}:\sum_{k=1}^{L} D_k = D_T} \sum_{k=1}^{L} \left[R_{X_k}^{(SL)}(D_k) \right] \\
&= \sum_{k=1}^{L} R_{X_k}^{(SL)}(D_k^*)
\end{aligned} \tag{3.18}$$

where

$$D_k^* = \begin{cases} d & \text{if } d \le \hat{\sigma}_k^2 \\ \hat{\sigma}_k^2 & \text{otherwise} \end{cases} \tag{3.19}$$

and d is chosen such that

$$\sum_{k=1}^{L} D_k^* = D_T. \tag{3.20}$$

The solution D_k^*, $1 \le k \le L$, can be interpreted as a kind of reverse water-filling[4], as illustrated in Fig.3.2. Comparing D_k^* in (3.19) and (3.20) with $D_k(Q)$ in (3.15) and (3.16),

[4]This reverse water-filling result is similar to the optimal distortion allocation for parallel Gaussian sources, for which Shannon lower bound on rate distortion functions is tight [12].

we define $Q^* = (q_1^*, q_2^*, \cdots, q_L^*)$ such that

$$q_k^* = \sup\{q_k : D(X_k, q_k) \le D_k^*\} \tag{3.21}$$

for any $1 \le k \le L$. In general, $D(X_k, q_k)$ is a strictly increasing and differentiable function of q_k, which will be assumed in our subsequent derivations. Then we have

$$D(X_k, q_k^*) = D_k^*, \tag{3.22}$$

which, together with (3.15), (3.16), (3.19) and (3.20), implies

$$D_k(Q^*) = D_k^*, \ 1 \le k \le L.$$

Indeed, among all Q satisfying (3.12) and $D_k(Q) = D_k^*$, $1 \le k \le L$, Q^* is the largest in the sense that

$$\delta(Q) \stackrel{\Delta}{=} \max\{q_k - q_k^* : 1 \le k \le L\} \le 0 \tag{3.23}$$

for any Q satisfying (3.12) and $D_k(Q) = D_k^*$, $1 \le k \le L$.

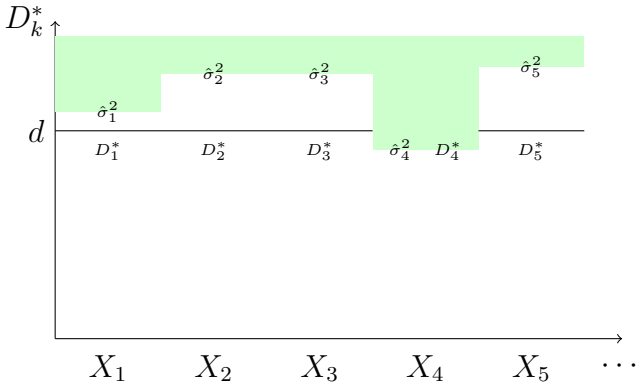

Figure 3.2: Reverse water-filling illustration for D_k^*.

Go back to (3.17). Note that the first summation in (3.17) is a non-decreasing function of q_k, $1 \le k \le L$, whereas the second summation in (3.17) is a strictly decreasing function

of q_k. Nonetheless, when n is large enough, the first summation is dominating. Therefore, an optimal $Q^o = (q_1^o, q_2^o, \cdots, q_L^o)$ to (3.17) would be the Q which tries to first minimize the first summation in (3.17) and then the second summation in (3.17) if there is room. In other words, we would expect that the optimal Q^o is either equal to Q^* or very close to Q^*. We next make this argument rigorous.

For any Q satisfying (3.12), define

$$F(Q) \stackrel{\Delta}{=} \sum_{k=1}^{L} R_{X_k}^{(SL)}(D_k(Q)) + \sum_{k=1}^{L} \left(\frac{c}{q_k} \frac{\ln n}{n} + o(\frac{\ln n}{n}) \right)$$

and

$$D(Q) \stackrel{\Delta}{=} (D_1(Q), D_2(Q), \cdots, D_L(Q)).$$

We want to compare $F(Q^*)$ with $F(Q)$. We distinguish among different cases: (i) $\delta(Q) < 0$; (ii) $\delta(Q) \geq C_1\sqrt{\frac{\ln n}{n}}$; (iii) $C_2\frac{\ln n}{n} \leq \delta(Q) < C_1\sqrt{\frac{\ln n}{n}}$; and (iv) $0 \leq \delta(Q) < C_2\frac{\ln n}{n}$, where $C_1 > 0$ and $C_2 > 0$ are constants to be specified later. In Case (i), we have

$$D_k(Q) = D_k^* = D_k(Q^*),\ 1 \leq k \leq L$$

and hence $F(Q) > F(Q^*)$. In Case (ii), in view of the strictly increasing and differentiable property assumption about $D(X_k, q_k)$, it follows from (3.15), (3.16), (3.19) and (3.20) that $|D(Q) - D(Q^*)| \geq c_1 C_1 \sqrt{\frac{\ln n}{n}}$ for some constant $c_1 > 0$, where $|D(Q) - D(Q^*)|$ denotes the Euclidean distance between $D(Q)$ and $D(Q^*)$. This, together with the convexity of $R_{X_k}^{(SL)}(D_k)$ and the optimality of D_k^*, implies that

$$\sum_{k=1}^{L} R_{X_k}^{(SL)}(D_k(Q)) > \sum_{k=1}^{L} R_{X_k}^{(SL)}(D_k(Q^*)) + \hat{c}_1 c_1^2 C_1^2 \frac{\ln n}{n}$$

for some constant $\hat{c}_1 > 0$, and hence

$$\begin{aligned} & F(Q) - F(Q^*) \\ > \; & \hat{c}_1 c_1^2 C_1^2 \frac{\ln n}{n} + \sum_{k=1}^{L} \frac{c}{q_k} \frac{\ln n}{n} - \sum_{k=1}^{L} \frac{c}{q_k^*} \frac{\ln n}{n} + o(\frac{\ln n}{n}) \\ > \; & \hat{c}_1 c_1^2 C_1^2 \frac{\ln n}{n} - \sum_{k: q_k > q_k^*} \frac{c}{q_k^*} \frac{\ln n}{n} + o(\frac{\ln n}{n}) \\ > \; & 0 \end{aligned}$$

for large n, when $C_1 > 0$ is chosen properly. In Case (iii), a similar argument can be used to show that

$$\begin{aligned} & F(Q) - F(Q^*) \\ > \; & \hat{c}_1 |D(Q) - D(Q^*)|^2 \\ & \quad - \sum_{k: q_k > q_k^*} \left(\frac{c}{q_k^*} - \frac{c}{q_k} \right) \frac{\ln n}{n} + o(\frac{\ln n}{n}) \\ > \; & c_3 \delta(Q) \left[c_2 |D(Q) - D(Q^*)| - \frac{\ln n}{n} \right] + o(\frac{\ln n}{n}) \\ \geq \; & c_3 \delta(Q) \left[c_2 c_1 C_2 \frac{\ln n}{n} - \frac{\ln n}{n} \right] + o(\frac{\ln n}{n}) \\ > \; & 0 \end{aligned}$$

for some constants $c_2 > 0, c_3 > 0$, and large n, when $C_2 > 0$ is properly chosen. Therefore, the optimal $Q^o = (q_1^o, q_2^o, \cdots, q_L^o)$ to (3.17) falls into Case (iv), i.e.,

$$0 \leq \delta(Q^o) < C_2 \frac{\ln n}{n},$$

which, together with (3.15), (3.16), (3.19) and (3.20), further implies that $q_k^o \geq q_k^* + O(\frac{\ln n}{n})$ for any $q_k^o < q_k^*$, and hence

$$|Q^o - Q^*| = O(\frac{\ln n}{n}).$$

To summarize, we have proved the following result.

Theorem 4 *Assume that $D(X_k, q_k)$, $1 \leq k \leq L$, is a strictly increasing and differentiable function of q_k. Then the optimal $Q^o = (q_1^o, q_2^o, \cdots, q_L^o)$ to (3.13) satisfies*

$$|Q^o - Q^*| = O(\frac{\ln n}{n})$$

where $Q^* = (q_1^*, q_2^*, \cdots, q_L^*)$ is defined through (3.19), (3.20), and (3.22).

Applying Theorem 4 in practice, we could simply take Q^* as Q^o since n is generally large. To further simplify the computation of Q^* without involving the computation of differential entropy, we could also replace $\hat{\sigma}_k^2$ by σ_k^2 in (3.19) and (3.20). In view of (3.8) and (3.11), it is reasonably believed that such Q^* would be a good approximation to the optimal Q in the original problem (3.4), especially in the case of high rate (or equivalently small D_T) coding, where the Shannon lower bound is quite close to the actual Shannon rate distortion function (see [79], [26], and references therein for the tightness of the Shannon lower bound). In addition, the determination of Q^* via (3.19), (3.20), and (3.22) is independent of any specific lossless coding method, and could be applied, in principle, to any practical image/video coding systems where quantization table design is involved. In the next section, we will apply Theorem 4 to some DCT-based image coding systems.

3.4 Application to DCT-based Image Compression

In JPEG and some DCT-based image coding such as [55] and [56], an image is first partitioned into non-overlapping 8×8-blocks, and each of the resulting 8×8-block is then transformed by a 8×8 DCT transform. Thus, in this case we have $L = 64$.

To apply Theorem 4 to DCT-based image coding, there are two issues we need to look at: 1) how to model DCT coefficients at each frequency i, $1 \leq k \leq 64$, and 2)

how to calculate $D(X_k, q_k)$ and q_k^* from (3.22) or equivalently (3.21). For DC coefficients, which are corresponding to $k = 1$, it is hard to find a proper probability model because of their irregularity. As such, we would simply model them as a uniformly distributed random source $X_1 = \{X_1(i)\}_{i=1}^n$. Accordingly, we have $D(X_1, q_1) = q_1^2/12$, where q_1 is the quantization step size for DC coefficients.

For AC coefficients, several good models have been proposed in the literature as shown in Chapter 2. Due to their advantage of having good trade-off between modelling accuracy and parameter estimation complexity, we shall focus only on the Laplacian model in what follows in this chapter.

According to the Laplacian distribution, AC coefficients $X_k = \{X_k(i)\}_{i=1}^n$ at each frequency $2 \leq k \leq 64$ are modelled with pdf

$$f(x_k) = \frac{1}{2\lambda_k} e^{-\frac{|x_k|}{\lambda_k}}, \ 2 \leq k \leq 64, \tag{3.24}$$

where $\lambda_k > 0$ is called a scale parameter at the ith frequency position in zig-zag order, and can be estimated, in practice, from the sample values $x_k(1), x_k(i), \cdots, x_k(n)$ of X_k, where $n = W \times H/64$, and W and H denote the width and height of the image to be encoded, respectively. In particular, the ML estimation of λ_k is calculated as follows

$$\lambda_k = \frac{1}{n} \sum_{k=1}^{n} | \ x_{i,k} \ |, \ 2 \leq k \leq 64. \tag{3.25}$$

With (3.24) and (3.25) in place, $D(X_k, q_k)$ can be calculated easily, and so is q_k^* with $D_k^* = D(X_k, q_k^*)$ in (3.22). Note, however, that $D(X_k, q_k)$ is the smallest distortion achievable with quantization step size q_k for the source X_k, which is given by the uniform quantizers with step size q_k. If q_k^* with $D_k^* = D(X_k, q_k^*)$ was chosen, then the uniform quantizers with step size q_k^* would have to be used to achieve D_k^*, which leaves no room for the subsequent RD trade-off with the reconstruction space $\{0, \pm 1q_k^*, \pm 2q_k^*, \cdots, \pm\lfloor 1 + \frac{A}{q_k^*} \rfloor q_k^*\}^n$. To overcome

this problem, we will reduce the value of q_k^* slightly by using a distortion function different from $D(X_k, q_k)$ in (3.21) and (3.22). Specifically, instead of using $D(X_k, q_k)$ in (3.21) and (3.22), we will use the distortion of a dead-zone quantizers with uniform reconstruction [50], [52]. Given the uniform reconstruction rule in JPEG and a quantization step size q_k, $2 \leq k \leq 64$, the corresponding dead-zone size (for the positive part) s_k, is computed by [52]

$$s_k = q_k - \lambda_k + \frac{q_k}{e^{q_k/\lambda_k} - 1}. \tag{3.26}$$

Let $D_{Lap}(\lambda_k, q_k)$ denote the distortion of the resulting dead-zone quantizers with the quantization step size q_k for the Laplacian source X_k with the scale parameter λ_k, $2 \leq k \leq 64$. It can be shown (after simplifying eq. (9) in [52]) that $D_{Lap}(\lambda_k, q_k)$ can be computed as follows

$$D_{Lap}(\lambda_k, q_k) = 2\lambda_k^2 - \frac{2q_k(\lambda_k + s_k - 0.5q_k)}{e^{s_k/\lambda_k}(1 - e^{-q_k/\lambda_k})}. \tag{3.27}$$

To compute our desired q_k^*, we then use $D_{Lap}(\lambda_k, q_k)$ in place of $D(X_k, q_k)$ in (3.21) and (3.22).

Based on the Laplacian model (3.24) and (3.25), our desired $Q^* = (q_1^*, q_2^*, \cdots, q_{64}^*)$ can be determined as follows. Predetermine a maximum integer quantization step size q_{max}. If the water level d in (3.19) and (3.20) is greater than the source variance σ_k^2, $k = 1, 2, \cdots, 64$, we directly quantize all coefficients at this frequency to zeros. This strategy is called fast quantization. We then set $q_k^* = q_{max}$ (note that, in this case, q_k^* is dummy when HDQ is considered since no matter what q_k^* is, the corresponding reconstruction level is always 0; however, this is not true in general for SDQ, as the quantized zero is the initialization for the iterative algorithm). Otherwise, q_k^* is selected such that

$$q_k^* = \begin{cases} \min\{\lfloor\sqrt{12d}\rfloor, q_{max}\} & \text{if } i = 1 \\ \max\{q_k \in \mathcal{Q} : D_{Lap}(\lambda_k, q_k) \leq d\} & \text{if } 2 \leq k \leq 64 \end{cases} \tag{3.28}$$

where $\mathcal{Q} = \{1, 2, \cdots, q_{max}\}$. The maximization problem in (3.28) can be solved by the bi-section search over $\mathcal{Q}$ for $2 \leq k \leq 64$. The procedure is summarized in Algorithm 4.

Algorithm 4 Optimal quantization table design for JPEG type DCT-based encoding based on the Laplacian model.

1: Predetermine a desired distortion level D_T per 8×8-block and a maximum quantization step size q_{max}.
2: Determine the water level d according to (3.19) and (3.20).
3: **if** $\sigma_k^2 < d$ **then**
4: set $q_k^* = q_{max}$
5: **else**
6: **if** $k = 1$ **then**
7: set $q_k^* = \min\{\lfloor\sqrt{12d}\rfloor, q_{max}\}$
8: **else**
9: set $q_k^* = \max\{q_k \in \mathcal{Q} : D_{Lap}(\lambda_k, q_k) \leq d\}$.
10: **end if**
11: **end if**

3.5 Experimental results

Having described our quantization table design algorithm based on the Laplacian model, i.e., Algorithm 4, we now evaluate its performance first in baseline JPEG encoding and then in ARL [55] and ECEB [56]. Experiments have been conducted on a set of standard 8-bit gray scale test images with different resolutions. In all experiments, q_{max} is set as 46.

In JPEG cases, customized Huffman tables are used. Both the HDQ coding (i.e.,

SDQ off) and SDQ coding (i.e., SDQ on) have been tested. When SDQ is on, the iterative algorithm in [72] has been further applied to provide a complete solution to the optimization problem (3.3). To facilitate our subsequent discussion, we shall refer to HDQ coding with quantization table designed by Algorithm 4 as J-OptD-HDQ, and SDQ coding with its initial quantization table designed by Algorithm 4 as J-OptD-SDQ. Tables 3.1-3.6 show the PSNR performance of J-OptD-HDQ and J-OptD-SDQ for 512×512 'Airplane (F16)', 512×512 'GoldHill', 512×512 'Lena', 512×512 'Dome', 720p 'Stockholm' (the first frame of the corresponding testing video sequence in [32]), and 1080p 'Kimono' (the first frame of the corresponding testing video sequence in [32]) respectively. Also shown in Tables 3.1-3.6 are the PSNR performance of the HDQ coding with quantization table designed by methods in [19] and [42] (hereafter referred to as J-OptQ-HDQ and J-RDOPT-HDQ, respectively) and the PSNR performance of the SDQ coding with its initial quantization table designed by methods in [19] and [42] (hereafter referred to as J-OptQ-SDQ and J-RDOPT-SDQ, respectively). As mentioned earlier, before our present work, J-RDOPT-HDQ represents the state-of-the-art JPEG HDQ optimizer, and J-RDOPT-SDQ represents the state-of-the-art JPEG SDQ optimizer [72]. In addition, the PSNR performance of baseline JPEG coding using a (scaled) default quantization table is listed in Tables 3.1-3.6 as an anchor. On the other hand, Table 3.13 shows the computer running times[5] of different quantization table design schemes along with the running times of other JPEG encoding components for compressing some 512×512 images. In Table 3.13, J-OptQ, J-RDOPT, OptD represent the quantization table design methods in the references [19], [42], and our proposed Algorithm 4, respectively. The results of our proposed schemes are bolded in all the tables in this thesis.

As can be seen from these tables, the experimental results for all tested schemes are

[5] All experiments in this thesis were run on an Apple Mac Pro 8-core 2.4GHz 12G RAM computer.

highly consistent with our discussion in Section 3.1 and Section 3.4 regarding both coding performance and computational complexity. When SDQ is off, our proposed J-OptD-HDQ significantly outperforms J-OptQ-HDQ and J-RDOPT-HDQ by 0.7 dB[6] and 0.5 dB on average with complexity reduced by a factor of more than 150 and 2000, respectively. When SDQ is on, the proposed J-OptD-SDQ provides a 0.2 dB gain and a 0.1 dB gain or more on average over J-OptQ-SDQ and J-RDOPT-SDQ respectively, with about 30% and 85% complexity reduced accordingly. Compared with baseline JPEG, our proposed J-OptQ-HDQ offers an average 1.5 dB gain with negligible complexity increase; and our proposed J-OptQ-SDQ provides an average 2.0 dB gain or more with a slight increase in computational complexity (due to the SDQ algorithm). Another interesting observation is that in both HDQ coding and SDQ coding, the performance gain from our proposed quantization table design schemes, tends to be larger when the rate is higher, which is consistent with our statement in the last paragraph of Section 3.3. This, together with the significant overall coding gain, validates our theoretical findings in Section 3.3 once again.

The complexity overhead (on top of baseline JPEG) of J-OptD-HDQ mainly comes from the calculation of the variance σ_k^2 and the ML estimate of λ_k. In some very low/high rate coding cases, this complexity can be further reduced by skipping the variance calculation for some high/low frequencies, as σ_k^2 can be always smaller/larger or equal to the water level d. For low rate encoding, sometimes the complexity introduced by J-OptD-HDQ can almost be compensated by the fast quantization strategy.

In what follows, we will discuss the related experiments on ARL and ECEB. In ARL cases [7], both the HDQ and SDQ coding have been tested. When SDQ is on, the SDQ algorithm proposed in [71] and its corresponding iterative algorithm has been applied. The

[6] Generally, a 0.1 dB gain in PSNR is equivalent to 2-3% bit rate reduction in JPEG encoding.

[7] All ARL related tests are run on our implementation of [55].

HDQ coding with quantization table designed by Algorithm 4 is dubbed A-OptD-HDQ, and SDQ coding with its initial quantization table designed by Algorithm 4 is referred to as A-OptD-SDQ. Tables 3.7-3.12 show the PSNR performance of A-OptD-HDQ and A-OptD-SDQ for all the test images in the same order as the JPEG cases, which are compared with those of the ARL HDQ and SDQ coding with a (scaled) default JPEG quantization table (hereafter referred to as A-DefQ-HDQ and A-DefQ-SDQ, respectively). For brevity's sake, the ECEB[8] HDQ PSNR performance (no ECEB SDQ coding performance is shown, since there is no SDQ algorithm designed for ECEB) with a (scaled) default JPEG quantization table, a uniform quantization table (as in the original ECEB codec, i.e., all quantization step sizes in the quantization table are the same), and a quantization table designed by Algorithm 4 are demonstrated in Tables 3.7-3.12 as well, referred to as E-DefQ-HDQ, E-UnifQ-HDQ and E-OptD-HDQ, respectively. Computer running times of all encoding components for the related ARL or/and ECEB tests are illustrated in Table 3.14. Again, OptD represents Algorithm 4 in Table 3.14.

On average, A-OptD-HDQ and A-OptD-SDQ significantly outperform their respective counterpart A-DefQ-HDQ and A-DefQ-SDQ by 1.4 dB and 0.8 dB in PSNR, and E-OptD-HDQ provides a 1.5 dB and a 0.4 dB notable performance gain over E-DefQ-HDQ and E-UnifQ-HDQ, respectively. According to Table 3.14, the complexity introduced by Algorithm 4 is negligible for both ARL and ECEB encoding.

[8]All ECEB related tests are run without the pre-processing and on the code kindly provided by the authors of [55] and [56].

Table 3.1: PSNR performance comparison of different Q-table design methods for baseline JPEG encoding for 512 × 512 Airplane (F16)

Rate	J-DefQ-HDQ	J-OptQ-HDQ	J-RDOPT-HDQ	J-OptD-HDQ	J-OptQ-SDQ	J-RDOPT-SDQ	J-OptD-SDQ
0.25	31.66	31.89	32.06	**32.16**	32.31	32.64	**32.70**
0.50	35.71	35.99	36.22	**36.30**	36.80	36.73	**36.87**
0.75	37.92	38.63	38.93	**39.06**	39.39	39.38	**39.56**
1.00	39.61	40.56	40.81	**41.14**	41.32	41.32	**41.47**
1.25	40.99	42.16	42.39	**42.81**	42.94	42.92	**43.11**
1.50	42.22	43.43	43.74	**44.16**	44.26	44.29	**44.49**
1.75	43.32	44.63	44.98	**45.45**	45.38	45.55	**45.77**
2.00	44.34	45.79	46.14	**46.72**	46.57	46.79	**46.98**

Table 3.2: PSNR performance comparison of different Q-table design methods for baseline JPEG encoding for 512 × 512 GoldHill

Rate	J-DefQ-HDQ	J-OptQ-HDQ	J-RDOPT-HDQ	J-OptD-HDQ	J-OptQ-SDQ	J-RDOPT-SDQ	J-OptD-SDQ
0.25	29.30	29.45	29.38	**29.63**	29.94	30.00	**29.96**
0.50	31.72	31.98	31.98	**32.30**	32.64	32.64	**32.65**
0.75	33.26	33.76	33.85	**34.25**	34.51	34.52	**34.56**
1.00	34.55	35.19	35.36	**35.88**	36.05	36.05	**36.07**
1.25	35.62	36.51	36.71	**37.27**	37.37	37.51	**37.54**
1.50	36.65	37.73	37.95	**38.55**	38.68	38.81	**38.88**
1.75	37.66	38.81	39.15	**39.73**	39.77	40.03	**40.08**
2.00	38.53	39.98	40.28	**40.93**	41.04	41.25	**41.29**

Table 3.3: PSNR performance comparison of different Q-table design methods for baseline JPEG encoding for 512 × 512 Lena

Rate	J-DefQ-HDQ	J-OptQ-HDQ	J-RDOPT-HDQ	J-OptD-HDQ	J-OptQ-SDQ	J-RDOPT-SDQ	J-OptD-SDQ
0.25	31.64	31.70	31.85	**31.89**	32.37	32.47	**32.41**
0.50	34.90	35.28	35.41	**35.53**	35.89	36.04	**36.01**
0.75	36.62	37.33	37.41	**37.77**	37.99	38.14	**38.16**
1.00	37.91	38.71	38.82	**39.31**	39.47	39.63	**39.66**
1.25	38.98	39.79	40.01	**40.52**	40.62	40.83	**40.90**
1.50	39.96	40.89	41.05	**41.69**	41.76	41.93	**41.97**
1.75	40.75	41.84	42.10	**42.71**	42.78	43.02	**43.10**
2.00	41.66	42.88	43.21	**43.84**	43.80	44.06	**44.12**

Table 3.4: PSNR performance comparison of different Q-table design methods for baseline JPEG encoding for 512 × 512 Dome

Rate	J-DefQ-HDQ	J-OptQ-HDQ	J-RDOPT-HDQ	J-OptD-HDQ	J-OptQ-SDQ	J-RDOPT-SDQ	J-OptD-SDQ
0.25	31.20	31.25	31.31	**31.43**	31.72	31.77	**31.72**
0.50	33.69	34.04	34.08	**34.30**	34.66	34.66	**34.69**
0.75	35.31	35.92	36.04	**36.35**	36.64	36.74	**36.76**
1.00	36.55	37.48	37.66	**38.07**	38.42	38.45	**38.52**
1.25	37.69	38.93	39.24	**39.83**	40.00	40.03	**40.07**
1.50	38.81	40.42	40.74	**41.37**	41.56	41.58	**41.59**
1.75	39.89	41.93	42.28	**42.88**	42.80	42.99	**43.21**
2.00	40.96	43.13	43.79	**44.31**	43.83	44.17	**44.53**

Table 3.5: PSNR performance comparison of different Q-table design methods for baseline JPEG encoding for 720p Stockholm (1st frame)

Rate	J-DefQ-HDQ	J-OptQ-HDQ	J-RDOPT-HDQ	J-OptD-HDQ	J-OptQ-SDQ	J-RDOPT-SDQ	J-OptD-SDQ
0.25	30.69	30.77	30.78	**31.03**	31.20	31.23	**31.27**
0.50	33.39	33.67	33.72	**34.04**	34.26	34.24	**34.27**
0.75	35.03	35.45	35.53	**35.96**	35.91	35.91	**36.10**
1.00	36.23	36.75	36.88	**37.40**	37.51	37.57	**37.66**
1.25	37.27	37.89	38.12	**38.66**	38.62	38.71	**38.88**
1.50	38.11	38.97	39.25	**39.82**	39.78	39.82	**40.05**
1.75	38.97	40.04	40.38	**40.96**	40.99	41.09	**41.18**
2.00	39.88	41.21	41.51	**42.16**	41.97	42.25	**42.33**

Table 3.6: PSNR performance comparison of different Q-table design methods for baseline JPEG encoding for 1080p Kimono (1st frame)

Rate	J-DefQ-HDQ	J-OptQ-HDQ	J-RDOPT-HDQ	J-OptD-HDQ	J-OptQ-SDQ	J-RDOPT-SDQ	J-OptD-SDQ
0.25	39.74	39.79	39.82	**40.02**	40.16	40.29	**40.37**
0.50	42.78	42.82	42.85	**43.03**	42.91	43.09	**43.11**
0.75	44.01	43.85	44.04	**44.35**	44.29	44.45	**44.47**
1.00	44.79	44.58	44.82	**45.36**	45.27	45.46	**45.52**
1.25	45.43	45.53	45.71	**46.32**	46.29	46.37	**46.57**
1.50	45.98	46.45	46.63	**47.41**	47.50	47.59	**47.65**
1.75	46.56	47.29	47.59	**48.42**	48.29	48.54	**48.62**
2.00	47.11	48.38	48.80	**49.31**	49.20	49.52	**49.60**

Table 3.7: PSNR performance comparison of different quantization table design methods for ARL and ECEB encoding for 512×512 Airplane (F16)

Rate	A-DefQ-HDQ	A-OptD-HDQ	A-DefQ-SDQ	A-OptD-SDQ	E-DefQ-HDQ	E-UnifQ-HDQ	E-OptD-HDQ
0.25	32.78	**32.98**	32.81	**33.04**	32.90	33.23	**33.31**
0.50	36.44	**36.86**	37.11	**37.36**	36.62	37.41	**37.20**
0.75	38.54	**39.68**	39.73	**39.99**	38.85	40.02	**40.21**
1.00	40.24	**41.71**	41.57	**42.01**	40.53	41.91	**42.11**
1.25	41.48	**43.28**	42.95	**43.61**	41.92	43.45	**43.86**
1.50	42.62	**44.74**	44.15	**45.03**	43.17	44.79	**45.33**
1.75	43.73	**46.02**	45.19	**46.25**	44.30	46.00	**46.77**
2.00	44.74	**47.24**	45.93	**47.55**	45.38	47.08	**47.97**

Table 3.8: PSNR performance comparison of different quantization table design methods for ARL and ECEB encoding for 512×512 GoldHill

Rate	A-DefQ-HDQ	A-OptD-HDQ	A-DefQ-SDQ	A-OptD-SDQ	E-DefQ-HDQ	E-UnifQ-HDQ	E-OptD-HDQ
0.25	29.93	**30.04**	29.99	**30.12**	29.88	30.09	**30.21**
0.50	32.35	**32.78**	32.67	**32.87**	32.34	32.84	**32.90**
0.75	33.88	**34.72**	34.60	**34.88**	33.94	34.77	**34.85**
1.00	35.16	**36.34**	35.72	**36.54**	35.33	36.39	**36.58**
1.25	36.21	**37.81**	36.91	**37.94**	36.52	37.87	**38.07**
1.50	37.35	**39.11**	38.05	**39.36**	37.65	39.21	**39.47**
1.75	38.27	**40.41**	39.26	**40.76**	38.72	40.55	**40.80**
2.00	39.23	**41.70**	39.84	**42.02**	39.77	41.85	**42.14**

Table 3.9: PSNR performance comparison of different quantization table design methods for ARL and ECEB encoding for 512×512 Lena

Rate	A-DefQ-HDQ	A-OptD-HDQ	A-DefQ-SDQ	A-OptD-SDQ	E-DefQ-HDQ	E-UnifQ-HDQ	E-OptD-HDQ
0.25	32.67	**32.80**	32.98	**33.01**	32.57	33.04	**32.96**
0.50	35.52	**36.03**	36.27	**36.48**	35.60	36.47	**36.32**
0.75	37.15	**38.13**	38.20	**38.47**	37.32	38.43	**38.46**
1.00	38.38	**39.65**	39.67	**39.98**	38.66	39.86	**40.03**
1.25	39.45	**40.92**	40.95	**41.31**	39.78	41.12	**41.35**
1.50	40.36	**42.07**	41.98	**42.49**	40.80	42.29	**42.62**
1.75	41.17	**43.23**	42.87	**43.59**	41.76	43.50	**43.85**
2.00	42.08	**44.46**	44.06	**44.92**	42.71	44.69	**45.09**

Table 3.10: PSNR performance comparison of different quantization table design methods for ARL and ECEB encoding for 512×512 Dome

Rate	A-DefQ-HDQ	A-OptD-HDQ	A-DefQ-SDQ	A-OptD-SDQ	E-DefQ-HDQ	E-UnifQ-HDQ	E-OptD-HDQ
0.25	31.96	**32.10**	32.05	**32.17**	31.90	32.06	**32.26**
0.50	34.35	**34.89**	34.96	**35.15**	34.41	34.92	**35.04**
0.75	35.90	**36.93**	36.71	**37.04**	35.99	37.06	**37.14**
1.00	37.13	**38.77**	38.18	**39.14**	37.36	38.88	**39.10**
1.25	38.35	**40.54**	39.67	**40.87**	38.68	40.58	**40.88**
1.50	39.44	**42.16**	41.04	**42.47**	39.94	42.16	**42.53**
1.75	40.60	**43.73**	42.48	**43.92**	41.25	43.60	**44.13**
2.00	41.74	**45.25**	43.67	**45.36**	42.53	44.96	**45.66**

Table 3.11: PSNR performance comparison of different quantization table design methods for ARL and ECEB encoding for 720p Stockholm (1st frame)

Rate	A-DefQ-HDQ	A-OptD-HDQ	A-DefQ-SDQ	A-OptD-SDQ	E-DefQ-HDQ	E-UnifQ-HDQ	E-OptD-HDQ
0.25	31.37	**31.55**	31.51	**31.61**	31.40	31.55	**31.62**
0.50	33.98	**34.42**	34.57	**34.62**	34.06	34.51	**34.60**
0.75	35.55	**36.30**	36.29	**36.48**	35.66	36.34	**36.52**
1.00	36.68	**37.78**	37.44	**37.94**	36.91	37.83	**38.09**
1.25	37.72	**39.12**	38.48	**39.28**	38.02	39.21	**39.48**
1.50	38.69	**40.36**	39.51	**40.51**	39.07	40.54	**40.83**
1.75	39.61	**41.62**	40.48	**41.77**	40.09	41.85	**42.14**
2.00	40.53	**42.89**	41.32	**43.00**	41.11	43.13	**43.51**

Table 3.12: PSNR performance comparison of different quantization table design methods for ARL and ECEB encoding for 1080p Kimono (1st frame)

Rate	A-DefQ-HDQ	A-OptD-HDQ	A-DefQ-SDQ	A-OptD-SDQ	E-DefQ-HDQ	E-UnifQ-HDQ	E-OptD-HDQ
0.25	40.58	**40.59**	40.67	**40.69**	40.57	40.65	**40.79**
0.50	43.14	**43.17**	43.41	**43.16**	43.07	42.96	**43.47**
0.75	44.36	**44.71**	44.53	**44.93**	44.09	44.15	**44.85**
1.00	45.16	**45.86**	45.48	**46.11**	44.80	45.23	**46.01**
1.25	45.83	**46.88**	46.11	**47.23**	45.44	46.29	**47.11**
1.50	46.39	**48.10**	46.91	**48.42**	46.14	47.30	**48.37**
1.75	47.12	**49.02**	47.72	**49.35**	46.98	48.27	**49.26**
2.00	48.28	**50.25**	48.90	**50.57**	47.93	49.15	**50.62**

Table 3.13: Computer running time (in milliseconds) of different quantization table design methods and other encoding components for baseline JPEG encoding for 512×512 images

J-OptQ	J-RDOPT	OptD	SDQ for JPEG	Float DCT / Integer DCT	HDQ & Huffman coding
92	1304	**0.6**	230	41 / 7	7

Table 3.14: Computer running time (in milliseconds) of all encoding components for ARL or/and ECEB for 512×512 images

OptD	SDQ for ARL	Float DCT / Integer DCT	HDQ & ARL entropy coding	HDQ & ECEB entropy coding
0.6	290	41 / 7	25	45

3.6 Chapter summary

In this chapter, quantization table design problem has been revisited from a new perspective where SDQ is considered. Unlike the traditional quantization table design where an actual encoding method is assumed, we design a quantization table for the purpose of reconstruction. An optimal distortion profile for designing quantization tables has been derived under some assumptions, which provides a generic solution for the quantization table design problem for image/video coding. Based on our theoretical result, we have then proposed an efficient algorithm using the Laplacian model to optimize quantization tables for DCT-based image coding. When tested over standard images for baseline JPEG encoding, our algorithm achieves the best compression performance when SDQ is both on and off, with almost no extra burden on complexity. As such, the proposed quantization table optimization algorithm, together with the SDQ algorithm in [72], shall be treated as a benchmark for evaluating future JPEG encoding algorithms. In addition, to further verify this generic algorithm for DCT-based image coding, we have also applied it to ARL [55] and ECEB [56], yielding significantly boosted coding efficiency in both cases.

Chapter 4

An Efficient DCT-based Image Compression System Based on Laplacian Transparent Composite Model

Having proposed an improved probability model for continuous DCT coefficients[1], i.e., the LPTCM, in Chapter 2, and an advanced quantization table optimization scheme, i.e., the OptD, in Chapter 3, we are now considering designing a new DCT-based non-predictive lossy image compression system based on them (as shown in Fig. 4.1). In this chapter, we first review some related lossy coding algorithms proposed in the literature in Section 4.1. Based on the LPTCM and the OptD, we then propose a novel quantizer called transparent

[1]Fast float 8×8 DCT [10] is used in this chapter as a compromise between signal de-correlation performance and complexity.

composite quantizer (TCQ) in Section 4.2, which consists of a uniform quantizer (UQ) and a new structured quantizer dubbed constrained dead-zone quantizer (CDZQ). Next, in Section 4.3, we introduce a new entropy coding method named transparent composite coding (TCC), followed by a SDQ algorithm to joint optimize the proposed CDZQ and TCC. The multiresolution capability of the proposed system is discussed in Section 4.5, and experimental results are given in Section 4.6. Finally, we summarize this chapter in Section 4.7.

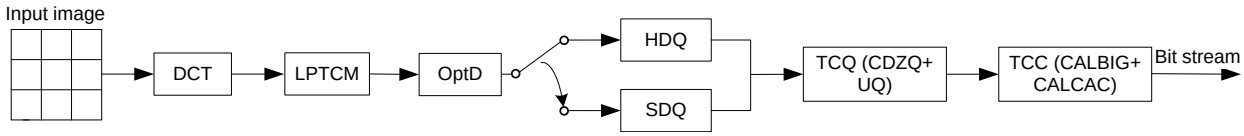

Figure 4.1: Block diagram of the proposed DCT-based non-predictive image compression system

4.1 Literature review

For DCT-based lossy image compression systems, knowledge of the probability distribution of DCT coefficients is important to quantization and entropy coding design. Laplacian has been a popular model in guiding quantizer design in the past two decades [50], [52]. Based on the Laplacian model, a so-called dead-zone quantizers (DZQ) as shown in (2.34) was proposed in [50], where the partition contains zero is generally larger than others, which are uniform, and reconstruction points are also uniformly calculated. DZQ is widely used in image/video coding because of its simplicity/practicality and relatively good compression performance. It is shown in [50] that the compression performance of a DZQ is close to that of an entropy constrained scaler quantizers (ECSQ), under the condition that the input source is strictly Laplacian. However, in real-world DCT-based image coding, the input

source to a quantizer is a sequence of DCT coefficients, which is modeled more accurately by the LPTCM rather than Laplacian, as shown in Chapter 2. Therefore, it is interesting to investigate quantizers and subsequent entropy coding design based on the LPTCM for DCT-based image compression. Other model-based lossy compression algorithms including [21], [19], [68], [69], [54], [53], etc.

As demonstrated in Chapter 2, the LPTCM not only achieves superior modelling accuracy, but also has a good capability of nonlinear data reduction by identifying and separating a DCT coefficient in the heavy tail (referred to as an outlier coefficient or simply outlier) from a DCT coefficient in the main body (referred to as an inlier coefficient or simply inlier). This in turn opens up opportunities for it to be used in DCT-based (lossy) image compression.

4.2 Transparent composite quantizers

Since the input source to the quantizers is modeled by a LPTCM instead of a Laplacian, the structure of a dead-zone quantizers (DZQ) is not suitable anymore. In this section, we propose a new hard-decision quantizers for AC coefficients called the TCQ based on the LPTCM. The TCQ is a *composite* quantizers, where a UQ is used to quantize outliers and a separate CDZQ to quantize inliers. It is also *transparent* in the sense that given the boundary parameter in the LPTCM, there is no ambiguity in deciding which quantizers to apply for the current encoding AC for both encoder and decoder (the boundary parameter will be transmitted to the decoder as well as other distribution and coding parameters as discussed later for the image header) side (similar reason applied for naming the TCC in Section 4.3).

The TCQ uses a quantization table, and to optimize the quantization table, the OptD

procedure proposed in Chapter 3 (i.e., (3.19) and (2.21)) is applied to decide which frequency location will be assigned a positive rate. The main idea of the OptD is to assign the same amount of distortion to each independent DCT coefficient source (coefficients at the same frequency form a source). If the source variance is greater than or equal to d, a pre-determined distortion level or water level [69], a positive rate is assigned to this source such that the resulting distortion is d. Otherwise, no bits are used to encode the frequency source, and all quantized indices at the corresponding frequency location are set to zeros directly, and thus quantization and entropy coding can actually be skipped in this case. As such, it makes our coding schemes more computationally efficient, especially in low-rate coding where quantization and entropy coding are possibly skipped for many frequency locations. In addition, the OptD technique can also potentially improve the coding efficiency as shown in [69]. To apply the OptD, we define a binary table called a termination table, or T table, $T = \{T_1, T_2, \cdots, T_{64}\}$, with its entries representing the OptD decision of termination or execution (entries equal to 1 or 0 accordingly) of the following quantization and entropy coding for the corresponding frequency locations in the zig-zag order, i.e., $T_k = 0$ if $\sigma_k^2 \geq d$, otherwise $T_k = 1$, where σ_k is the variance of the DCT coefficient source at the kth frequency location in the zig-zag order, for $1 \leq k \leq 64$.

We now consider the TCQ design based on the LPTCM proposed in Chapter 2, which provides better modelling accuracy than the Laplacian model while maintaining the simplicity and practicality similar to those of the latter. Given a sequence of AC coefficients $Y_k = \{Y_k(i)\}_{i=1}^n$, $2 \leq k \leq 64$, the LPTCM first separates the tail from the main body of the sequence, and then models the tail portion by a uniform distribution and the main body by a truncated Laplacian distribution. The pdf used to model $Y_k = \{Y_k(i)\}_{i=1}^n$ is given by

$$f(y_k) = \begin{cases} \frac{b_k}{2\lambda'_k(1-e^{-Y_k^c/\lambda'_k})} e^{-\frac{|y_k|}{\lambda'_k}} & \text{if } \mid y_k \mid < Y_k^c \\ \frac{1-b_k}{2(A_k - Y_k^c)} & \text{if } Y_k^c < |y_k| \leq A_k \\ 0 & \text{if } |y_k| > A_k \end{cases} \tag{4.1}$$

where, as we recall, $b_k \in [0,1]$, Y_k^c is the truncation point or separation boundary in the LPTCM at the ith frequency in a predefined scanning order, which separates the main body from the tail, λ'_k is the scale parameter of the truncated Laplacian distribution, and A_k denotes the largest possible amplitude a sample AC at the ith frequency can take. The ML estimates of Y_k^c, A_k, λ'_k, and b_k are computed through Algorithm 2 in Chapter 2. According to our experiments as mentioned in Chapter 2, b_k is usually close to 0.99, which means the probability of the outlier is only about 0.01. Although outliers are statistically insignificant, they, as demonstrated in Chapter 2, contain very important information, which reveals some unique global features of the original image. Because of their subjective and objective importance, outliers should be handled wisely and separately from the inliers. This will be a design guideline for the TCQ and TCC discussed in the following sections.

Since outliers are modeled in (4.1) by uniform distributions, it may be better off to quantize them separately according to a UQ. In the proposed TCQ, we use one UQ to quantize all outliers from all AC frequencies as well all DC coefficients (DCs). The quantization step size q_1 of the UQ is calculated by $\sqrt{d/12}$. q_1 is generally smaller than the quantization step size of the CDZQ (discussed later in this section), which is verified by our experiments. As such, q_1 is used in the UQ of the TCQ to quantize outliers, to protect them from suffering too much distortion. For each AC frequency $2 \leq k \leq 64$ (where $T_k = 0$), the UQ of the TCQ is given by

$$Q(y_k) = sign(y_k) \left(y_k + q_1 (round(\frac{||y_k| - Y_k^c + q_1/2|}{q_1}) - 0.5) \right)$$

for $y_k > Y_k^c$, and the corresponding quantized index is given by

$$c_k' = sign(y_k) \times (L_k + round(\frac{||y_k| - Y_k^c + q_1/2|}{q_1})), \tag{4.2}$$

where L_k denotes the largest possible value a quantized index for the inlier at the kth frequency (in a pre-defined scanning order) can take (L_k is actually one of the quantization parameters of CDZQ, and the determination of L_k is discussed later). A separate CDZQ is used to quantize all inliers at the AC frequency k. In what follows, we formulate the CDZQ problem in subsection 4.2.1, and solve it in subsection 4.2.2.

4.2.1 Constrained dead-zone quantizer design–Problem formulation

In this subsection, we study a new type of dead-zone quantizers called the constrained dead-zone quantizers based on the truncated Laplacian model. The reason why it is named *constrained* DZQ is that, its dead-zone size is constrained because of the truncation (in the LPTCM), i.e., the last quantization partition boundary (for the positive distributed part) must resides in Y_k^c, $2 \leq k \leq 64$. Suppose the partition rule is fixed as uniform partition (except for the one contains zero) as in the conventional dead-zone quantizers settings. We then have the following constraint

$$Y_k^c = u_k + L_k \cdot q_k, \text{ or equivalently } u_k = Y_k^c - L_k \cdot q_k, \tag{4.3}$$

where q_k and u_k denote the quantization step size and dead-zone size (for the positive distributed part) of the CDZQ for the kth frequency, respectively. Mathematically, the CDZQ of the TCQ is given by

$$Q(y_k) = \begin{cases} 0, \text{ if } |y_k| \leq u_k \\ sign(y_k) \left(\delta_k + q_k (round(\frac{||y_k| - u_k + \frac{q_k}{2}|}{q_k}) - 1) \right), \text{otherwise} \end{cases}$$

for $y_k \leq Y_k^c$, where the reconstruction $Q(y_k)$ is calculated as a centroid of each partition interval according to the truncated Laplacian distribution, with the reconstruction offset δ_k being the reconstruction point corresponding to the first positive partition interval, and

$$\delta_k = u_k + \lambda_k \ + q_k/(1 - e^{q_k/\lambda_k}). \tag{4.4}$$

The corresponding quantized index c_k is calculated by

$$c_k = sign(y_k) \times round(\frac{||y_k| - u_k + q_k/2|}{q_k}). \tag{4.5}$$

Based on the above CDZQ structure, given λ_k, Y_k^c and a target distortion d, our problem is to design a CDZQ by finding the optimal quantization step size $q_k^\star$ and dead-zone size $u_k^\star$, such that the RD cost is minimized, which is formulated as

$$\min_{u_k, q_k} R(\lambda_k, Y_k^c, u_k, q_k), \text{ s.t. } \begin{cases} D(\lambda_k, Y_k^c, u_k, q_k) = d \\ u_k = Y_k^c - L_k \cdot q_k \\ u_k \geq 0.5 \cdot q_k \\ L_k \in \mathbb{I}, q_k \in \mathbb{R} \end{cases} \tag{4.6}$$

where R and D represent the rate and distortion, respectively, which can be easily calculated given the quantizers structure and truncated Laplacian distribution parameters.

4.2.2 Constrained dead-zone quantizer design–Problem solution

Considering (4.3), u_k can be solely determined by L_k and q_k. Thus, given λ_k, Y_k^c and d, the CDZQ design problem is actually equivalent to finding the corresponding $L_k^\star$ and $q_k^\star$ in the RD sense. In practice, L_k falls into a close set of integers, i.e., $L_k \in \mathbb{L} = [0, L_{max}]$, and q_k can be quantized to a value from a smaller close set of real numbers $\mathbb{Q}$

$= \{q_{min}, \cdots, \text{MAX}\{q_{max}, Y_k^c/(L_k+0.5)\}\}$, rather than taking any possible values from the (positive) real line. We've found $L_{max} = 255$, q_{min}=1 and $q_{max} = 255$ to be efficient in our case (for quantizing AC coefficients). Then (4.6) can be rewritten as

$$\min_{L_k, q_k} R(\lambda_k, Y_k^c, L_k, q_k), \text{ s.t. } \begin{cases} D(\lambda_k, Y_k^c, L_k, q_k) = d \\ u_k = Y_k^c - L_k \cdot q_k \\ L_k \in \mathbb{L}, q_k \in \mathbb{Q}. \end{cases} \tag{4.7}$$

(4.7) can be further converted to an unconstrained optimization problem by the Lagrange multiplier method, which can then be solved using a fixed-slope lossy coding [81] algorithm that is summarized in Algorithm 5.

Algorithm 5 Fixed-slope lossy coding algorithm to find optimal CDZQ parameters (at slope θ) $(L_k^\star, q_k^\star)_\theta$, for $2 \le k \le 64$

1: Given λ_k, Y_k^c and d, initialize $\theta > 0$, $\theta_L < \theta$, $\theta_H > \theta$, $D(\lambda_k, Y_k^c, L_k^\star, q_k^\star)_\theta = \infty$, and $\epsilon > 0$.

2: **while** $\mid D(\lambda_k, Y_k^c, L_k^\star, q_k^\star)_\theta - d \mid > \epsilon$, **do**

$$(L_k^\star, q_k^\star)_\theta = \arg\min_{L_k, q_k} R(\lambda_k, Y_k^c, L_k, q_k) + \theta \cdot D(\lambda_k, Y_k^c, L_k, q_k);$$

3: **if** $D(\lambda_k, Y_k^c, L_k^\star, q_k^\star)_\theta < d$, set $\theta_H = \theta$, $\theta = (\theta + \theta_L)/2$;

4: **else** set $\theta_L = \theta$, $\theta = (\theta + \theta_H)/2$.

5: **end if**

6: **end while**

Remark 1: Algorithm 5 offers an optimal solution to (4.6), when $L_k \in \mathbb{L}$ and $q_k \in \mathbb{Q}$, for $2 \le k \le 64$.

In practice, Algorithm 5 can be performed in an off-line manner, and the optimal CDZQ

parameters can be stored in a look-up table shared by both the encoder and decoder. Since the decoder needs to know λ_k and Y_k^c to obtain the optimal CDZQ parameters, for $2 \leq k \leq 64$, their respective quantized versions $\hat{\lambda_k}$ and $\hat{Y_k^c}$ are used to replace λ_k and Y_k^c in the algorithm, and the corresponding quantized indices are transmitted to the decoder. Specifically, if $T_k = 0$, then we quantize λ_k into a discrete value, $\hat{\lambda_k}$, from a training set Λ, for $2 \leq k \leq 64$; define $\gamma_k \triangleq Y_k^c / \hat{\lambda_k}$ as the normalized boundary parameter, which is quantized to $\hat{\gamma_k}$ from a training set Γ. $\hat{Y_k^c}$ is calculated by $\hat{\lambda_k} \times \hat{\gamma_k}$. In this thesis, for $2 \leq k \leq 64$, λ_k and γ_k are uniformly quantized to one of 256 and 64 possible values respectively, with different quantization step sizes for different intervals according to the statistics of a test image set, and $\Lambda = \{.5 : .1 : 10,\ 10.2 : .2 : 20,\ 20.25 : .25 : 30,\ 30.5 : .5 : 40,\ 41 : 1 : 90\}$ and $\Gamma = \{1.25 : .25 : 3,\ 3.2 : .2 : 5,\ 5.1 : .1 : 8,\ 8.2 : .2 : 9,\ 9.25 : .25 : 11.75\}$, where the formats in the curly brackets between each comma are interpreted as "starting point: step size: end point". Since there are in total 256×64 combinations of $(\hat{\lambda},\ \hat{\gamma})$ pairs, the table has 16384 entries. For a given d, $L^\star$ and $q^\star$, or $\lfloor 100 \cdot q^\star \rfloor$[2] (for any AC frequencies), can be stored in the unsigned char and unsigned short data types, taking 1 byte and 2 bytes, respectively. So a CDZQ parameter table requires 48 KB of storage space, which is negligible considering the up-to-date device storage capacity. The quantized indices of $\hat{\lambda}$ and $\hat{\gamma}$ will be sent to the decoder. So the CDZQ parameters can be obtained simply by table-looking at both the encoder side and the decoder side. The quantized index c_k' or c_k is then encoded by the TCC, which is discussed in the next section.

[2] After table-looking, the entry is divided by 100 as quantization step size.

4.3 Transparent composite coding

In this section, we propose a novel entropy coding method called the transparent composite coding, to encode the quantized indices output by the TCQ. Prior to the TCC, an image header is written to the bit-stream. T table is sent first using 64 bits. For $2 \leq k \leq 64$, if $T_k = 0$, the corresponding quantized indices of $\hat{\lambda}_k$ and $\hat{\gamma}_k$ are then transmitted right after, taking 8 bits and 6 bits, respectively, which are followed by L'_k, taking another 8 bits, where $L'_k = round(\frac{A_k - Y_k^c + q_1/2}{q_1})$. In addition, the target distortion level d or the index of d is also needed to be included in the bit-stream.

According to the LPTCM or the TCQ, 1) the outlier and inlier represent two statistically different regions; 2) the probability of the appearance of an outlier is very small; 3) the quantized index of an outlier is larger in magnitude than that of an inlier at the same frequency. To take advantage of these observations, we propose to first encode and transmit the outlier flag (OTF), a binary information of whether the coefficient is an outlier ($OTF = 1$) or not ($OTF = 0$), and develop a bi-level image coding called context-adaptive layer-based bi-level image coding (CALBIC) to (losslessly) encode the OTF image, i.e., OTFs of all AC coefficients across the input image (left panel of Fig. 4.3 shows an example of an OTF image). In practice, the CALBIC can actually be applied right after the LPTCM algorithm and the OptD technique and before the TCQ, since the OTF image has already been available at that time. With the OTF image as side information, conditional quantization can be done such that the quantization process is modified by $c'_k = sign(y_k) \times round(\frac{||y_k| - Y_k^c + q_1/2|}{q_1})$ instead of (4.2), $2 \leq k \leq 64$, when the corresponding OTF is equal to 1. At the decoder side, with the knowledge of the OTF image, the de-quantization can be performed properly. We then conditionally encode the quantized indices of outliers and inliers separately by a composite entropy coding method

called context-adaptive layer-based composite arithmetic coding (CALCAC). Apart from offering chances to perform conditional quantization and conditional entropy coding, the OTF information is also beneficial in designing efficient high-order context models for the CALCAC. The OTF information based on the LPTCM allows separate quantization and entropy coding schemes for outliers and inliers. Note that our separation coding scheme that performs separation on a set of DCT coefficients is fundamentally different from those employed in JPEG lossless (JPEG-LS) [4], [60], [29], and JPEG-XR [49], where each encoding symbol is broken into significant information and a reminder, which are encoded using different coding schemes.

Another helpful side information is the alphabet size. In our LPTCM based coding scheme, alphabet sizes for the outlier and inlier at the kth frequency are L'_k and L_k+1, respectively, $2 \leq k \leq 64$, which are known by both of the encoder and decoder. Knowledge of the alphabet size of encoded symbols can improve entropy coding efficiency. It also leads to an adaptive scanning method as discussed later. To facilitate the rest discussion of the TCC, we define some notations as follows.

Block coordinate: a (8×8) block coordinate consists of two coordinates that solely decides its location in the input image in a 2-D manner, e.g., the block coordinate of the block in the mth row and nth column of the image is (m, n), $m \in [1, H/8]$, $n \in [1, W/8]$, where H and W denote the height and width of the image, respectively. Hereafter by mentioning block (m, n) we mean the block with the coordinate (m, n).

Frequency coordinate: similarly as a block coordinate, a frequency coordinate consists of two coordinates that solely decides its location in the encoding block, e.g., the frequency coordinate of the frequency at ith row and jth column of the current block is (i, j), $i \in [1, 8]$, $j \in [1, 8]$. Hereafter frequency (i, j) refers to the frequency with the coordinate (i, j)[3].

[3]Instead of the 1-D index k that we previously used to locate the frequency position, this 2-D coordinate

Scanning layer: a scanning layer (SL) is a group of AC frequencies that are on the same 45-degree diagonal line in a block; we say frequency (i_1, j_1) and frequency (i_2, j_2) are in the same SL if $i_1 + j_1 = i_2 + j_2$, i_1, i_2, j_1, $j_2 \in [1, 8]$, where the summation of the two coordinates is called a scanning layer index (SLI). A SL is the minimum coding unit in the TCC.

Context layer: a context layer (CL) is a group of AC frequencies whose contexts are merged together, as shown in Fig. 4.6, where there are in total 7 context layers, with each number at the corresponding frequency location representing a context layer index (CLI).

Block neighbour: a block neighbour (BN) (of the current encoding coefficient) is a coefficient at an adjacent frequency in the same block.

Frequency neighbour: a frequency neighbour (FN) is a coefficient at the same frequency in an adjacent block.

Scanning layer neighbour: a SL neighbour (SLN) is a coefficient at the same SL in the same block.

In the TCC, blocks are processed in the raster scan order, and elements in a block are processed by a layer-based adaptive scanning (LAS) method one SL by another, starting from the one with the greatest possible SLI[4]. As such, a SL can be seen as the minimum coding unit in the TCC. Within one SL, the frequency location with the smallest alphabet size[5] is scanned first followed by the second smallest and then carry on until all frequencies in this SL are scanned. Fig. 4.2 shows an example of LAS for a 8×8 DCT block with the entry rep-

will be used in this section for better assisting our presentation of the TCC.

[4]Again, only frequencies assigned positive rates by the OptD technique are scanned and entropy encoded.

[5]In the CALBIC, $L'_{(i,j)}$ is treated as the alphabet size information; in the CALCAC, depending on encoding of outliers or inliers, $L'_{(i,j)}$ or $L_{(i,j)} + 1$ is used as the alphabet size information, $i, j \in [1, 8]$.

resenting the corresponding L' of each frequency location. The adaptive scanning order is as follows shown by frequency co-ordinators in a first to last order with curly braces as the delimiters for the scanning layers – $\{(2,6)\}$, $\{(5,2),(1,6),(4,3)\}$, $\{(5,1),(2,4),(1,5),(3,3)\}$, $\{(2,3),(4,1),(3,2),(1,4)\}$,$\{(3,1),(1,3),(2,2)\}$,$\{(2,1),(1,2)\}$. In the following subsections, we discuss the CALBIC and CALCAC in detail.

	8	5	5	2	1	0	0
7	6	3	1	0	1	0	0
4	5	3	0	0	0	0	0
4	0	2	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 4.2: An example of LAS

4.3.1 Context-adaptive layer-based bi-level image coding

Since a bi-level OTF image is produced in DCT domain, strong data dependency can be found both within a (8×8) block (i.e., among BNs; see Fig 4.3: Left) and at the same frequencies among the adjacent blocks (i.e., among FNs; see right panel of Fig. 4.3). Therefore, in this case, traditional bi-level image coding methods (such as JBIG [2] and JBIG2 [3]) that exploit data dependency among pixels may not be competent. To efficiently compress the bi-level images, we propose a new entropy coding method named CALBIC in this subsection using binary arithmetic coding. High-order context modelling is designed to exploit correlation of the bins among the BNs and FNs.

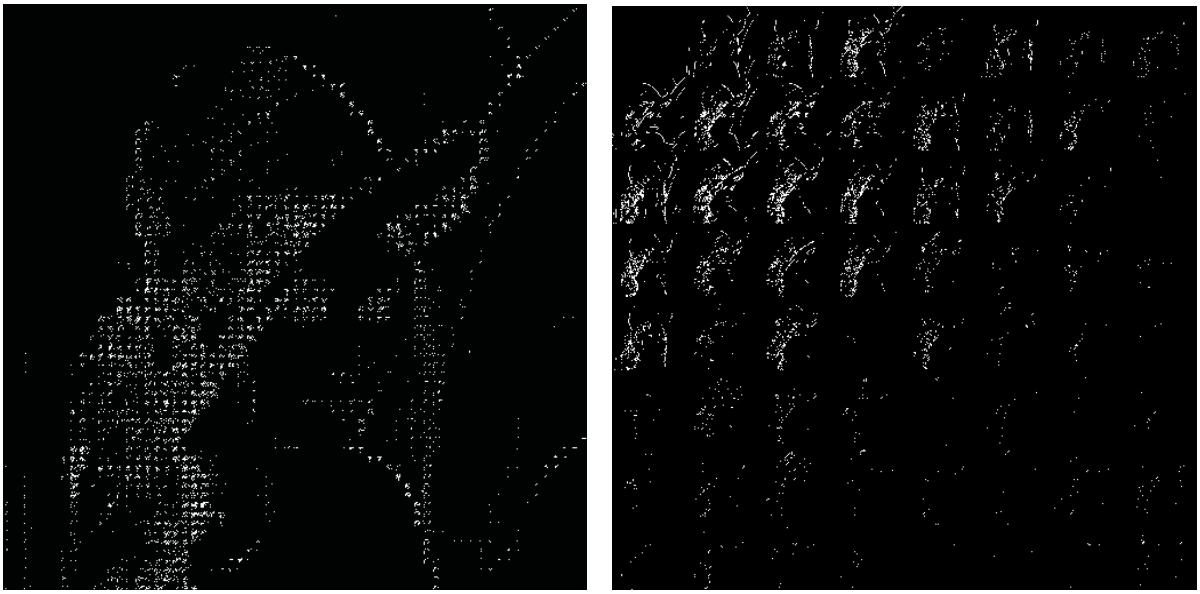

Figure 4.3: Left: OTF image of 512×512 Lena (when $d = 42$ dB in PSNR); Right: 63 sub-images of OTFs of 512×512 Lena in a frequency-by-frequency view (when $d = 42$ dB in PSNR)

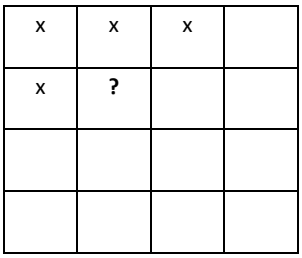

Figure 4.4: Context template for encoding OBF/ $SGN1$/ $SGN2$

To encode OTFs in the block (m, n), $m \in [1, H/8]$, $n \in [1, W/8]$, a binary information called an outlier block flag (OBF) is first encoded to specify if this block has at least one OTF appears ($OBF_{(m,n)} = 1$) or not ($OBF_{(m,n)} = 0$) (see Fig. 4.10: Top-middle for an example of an OBF image, i.e., OBFs for the whole image). If $OBF_{(m,n)} = 1$, we further encode OTFs in this block, or we stop the encoding of the current block and directly

go to the next one. This procedure continues until the last block in the input image is encoded. $OBF_{(m,n)}$ is coded based on the already encoded OBFs of its BNs (if any), with the context template shown in Fig. 4.4, where '?' is the current encoding OBF, and '$\times$'s denote the already encoded the OBF information of its FNs (if any). The corresponding context index increment $OBF\_inc_{(m,n)}$ is calculated by

$$\begin{aligned} OBF\_inc_{(m,n)} = OBF_{(m-1,n-1)} + 2 \cdot OBF_{(m-1,n)} \\ +4 \cdot OBF_{(m-1,n+1)} + 8 \cdot OBF_{(m,n-1)}. \end{aligned}$$

So there are 16 probability models reserved for encoding OBF. When $OBF_{(m,n)} = 1$ and $L'_{(i,j)} > 0$, the OTF at the frequency (i, j) of the block (m, n), $OTF_{(m,n),(i,j)}$, is further encoded according to the LAS order and based on the context template shown in Fig. 4.5, where '?' is the current encoding OTF, and 'o' corresponds to the OTF information of the immediate coded SLN[6] (if any), and '$\times$'s denote the already encoded the OTF information of its BNs or FNs (if any). The corresponding context index increment $OTF\_inc_{(m,n),(i,j)}$ is given in (4.8), where $m \in [1, H/8]$, $n \in [1, W/8]$, i, j, i', $j'$$\in [1, 8]$, and $OTF_{(m,n),(i',j')}$ denotes the OTF information of the immediate coded SLN of $OTF_{(m,n),(i,j)}$ at the frequency (i', j') (if any). $OTF\_inc_{(m,n),(i,j)}$ is truncated at 7. The contexts of the frequencies in the same context layer (see Fig. 4.6) are merged to avoid context dilution. So up to 56 probability models reserved for encoding OTF.

$$\begin{aligned} OTF\_inc_{(m,n),(i,j)} = OTF_{(m-2,n),(i,j)} + OTF_{(m-1,n-1),(i,j)} + OTF_{(m-1,n),(i,j)} + OTF_{(m-1,n+1),(i,j)} \\ +OTF_{(m+1,n-2),(i,j)} + OTF_{(m+1,n-1),(i,j)} + OTF_{(m,n),(i,j+1)} + OTF_{(m,n),(i,j+2)} \\ +OTF_{(m,n),(i+1,j)} + OTF_{(m,n),(i+1,j+1)} + OTF_{(m,n),(i+2,j)} + OTF_{(m,n),(i',j')}. \end{aligned} \tag{4.8}$$

[6]A immediate coded SLN is a SLN, which is scanned by LAS and encoded right before the current encoding coefficient.

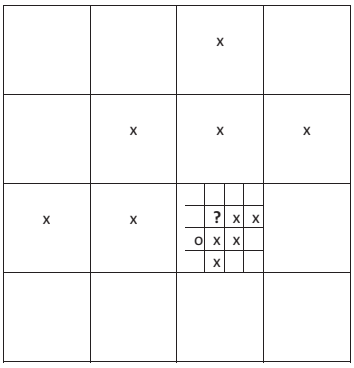

Figure 4.5: Context template for encoding OTF/ $OTL1$/ $OTL2$/ $ISIG0$/ $ISIG1$/ $ISIG2$

	1	3	4	5	5	6	6
2	3	4	5	5	6	6	6
3	4	5	5	6	6	6	7
4	5	5	6	6	6	7	7
5	5	6	6	6	7	7	7
5	6	6	6	7	7	7	7
6	6	6	7	7	7	7	7
6	6	7	7	7	7	7	7

Figure 4.6: Context layer indices

4.3.2 Context-adaptive layer-based composite arithmetic coding

With the OTF image as side information, the CALCAC handles the encoding of quantized indices including those of DCs and AC coefficients. For DCs, the energy of the quantized indices is very large, so it is worthwhile to use a predictor as JPEG does [1]. In the CALCAC, a quantized index for DC is first predicted by the predictor proposed in [63],

and then entropy coded by the multi-symbol (or m-array) arithmetic coding, to improve the low throughput problem seen in binary arithmetic coding based codec, such as ECEB [56], H.264/AVC [31] and HEVC [32]. For AC coefficients, the quantized indices in the block are first scanned by LAS, and then encoded by the procedure discussed in the next paragraph. Quantized indices of outliers and inliers are encoded separately based on different sets of context models that are well designed to capture high-order data dependencies.

Suppose we are currently encoding the block (m, n), there are two cases of $OBF_{(m,n)}$– either being 1 or 0, $m \in [1, H/8]$, $n \in [1, W/8]$. If the first case happens, we encode the quantized indices (including signs and magnitudes) of outliers first and then those of inliers in their respective LAS orders. For brevity's sake, we assume the indices to be nonnegative for now, and discuss the encoding of magnitudes first, and then cover that of sign bits later in the end of this subsection. By examining the statistics, we have found that most of the magnitudes of the quantized indices of outliers are 1 or 2, and those of inliers are 0, 1, or 2. As such, a significant-bit-coding-alike scheme [31], [32] is applied as follows. Provided $OTF_{(m,n),(i,j)}= 1$ and $L'_{(i,j)} > 1$, to encode the quantized index at the frequency (i, j) of the block (m, n), $c'_{(m,n),(i,j)}$, $m \in [1, H/8]$, $n \in [1, W/8]$, $i,\ j,\ \in [1, 8]$, a binary information $OTL1$ is encoded to indicate if $c'_{(m,n),(i,j)}$ is greater than 1 ($OTL1_{(m,n),(i,j)}= 1$) or not ($OTL1_{(m,n),(i,j)}= 0$). If $OTL1_{(m,n),(i,j)}= 1$ and $L'_{(i,j)} > 2$, another binary information $OTL2$ is encoded to signal if $c'_{(m,n),(i,j)}$ is greater than 2 ($OTL2_{(m,n),(i,j)}= 1$) or not ($OTL2_{(m,n),(i,j)}= 0$). $OTL1$ and $OTL2$ are encoded conditioned on their respective already coded information. Their context models are similar to OTF in a way that they share the same context template as in Fig. 4.5 and the same layer merging technique according to Fig. 4.6. The only difference is that the summations in (4.8)[7] are

[7]In this case, OTFs in (4.8) are substituted by $OTL1$s or $OTL2$s depending on which information is currently encoded.

truncated at 5 instead of 7 for both cases. The contexts for encoding $OTL1$ and $OTL2$ are merged together. Therefore, the total number of probability models for encoding $OTL1$ and $OTL2$ is up to 42. If $OTL2_{(m,n),(i,j)}= 1$ and $L'_{(i,j)} > 3$, $c'_{(m,n),(i,j)} - 3$ is further encoded, where 7 context models is used with the context index increments shown in Fig. 4.6.

In the case when $OTF_{(m,n),(i,j)}= 0$ and $L_{(i,j)} > 0$, a significant bit information $OSIG0$ is encoded to indicate if the corresponding index $c_{(m,n),(i,j)}$ is zero ($OSIG0_{(m,n),(i,j)}= 0$) or not ($OSIG0_{(m,n),(i,j)}= 1$). If $OSIG0_{(m,n),(i,j)}= 1$ and $L_{(i,j)} > 1$, a binary information $OSIG2$ is encoded to indicate if $c_{(m,n),(i,j)}$ is greater than 1 ($OSIG1_{(m,n),(i,j)}= 1$) or not ($OSIG1_{(m,n),(i,j)}= 0$). If $OSIG1_{(m,n),(i,j)}= 1$ and $L_{(i,j)} > 2$, another binary information $OSIG2$ is encoded to signal if $c_{(m,n),(i,j)}$ is greater than 2 ($OSIG2_{(m,n),(i,j)}= 1$) or not ($OSIG2_{(m,n),(i,j)}= 0$). $OSIG0$, $OSIG1$ and $OSIG2$ are encoded conditioned on their respective already coded information based on the context template shown in Fig. 4.7, where '?' is the current encoding information either being $OSIG0$ or $OSIG1$ or $OSIG2$, and, depending on the alphabet sizes and the LAS, one of 'o's corresponds to the corresponding information of its immediate coded SLN, and the other corresponds to the OTF of the immediate coded SLN whose corresponding L' is larger or equal to that of the current encoding frequency (if any), and 'x's denote the already encoded $OSIG0$s or $OSIG1$s or $OSIG2$s of its BNs or FNs (if any). The corresponding context index increment $OSIG0\_inc_{(m,n),(i,j)}$ for encoding the $OSIG0$ information is given in (4.9), where $m \in [1, H/8]$, $n \in [1, W/8]$, i, j, i', $j'\in [1, 8]$, $OSIG0_{(m,n),(i',j')}$ denotes the $OSIG0$ information of the immediate coded SLN of $OSIG0_{(m,n),(i,j)}$ at the frequency (i', j') (if any), and $OTF_{(m,n),(i'',j'')}$ denotes the OTF information of the closest SLN of $OSIG0_{(m,n),(i,j)}$ at the frequency (i'', j'') (if any), whose $OSIG0$ has not been encoded yet. $OSIG0\_inc_{(m,n),(i,j)}$ is truncated at 10. The contexts of the frequencies in the same context layer (see Fig. 4.6) are merged. Therefore, the number of probability models for encoding $OSIG0$ is up to 77. $OSIG1$ and $OSIG2$ are

encoded conditioned on their respective already coded information. Their context models are similar to $OSIG0$ in a way that they share the same context template as in Fig. 4.7 and the same layer merging technique according to Fig. 4.6. The only difference is that the summations in (4.9)[8] are truncated at 8 instead of 10 for both cases. The contexts for encoding $OSIG1$ and $OSIG2$ are merged together. So the total number of probability models for encoding $OSIG1$ and $OSIG2$ is up to 63. If $OSIG2_{(m,n),(i,j)}= 1$ and $L_{(i,j)} > 3$, $c_{(m,n),(i,j)} - 3$ is further encoded, where 7 context models is involved with the context index increments shown in Fig. 4.6.

$$
\begin{aligned}
&OSIG0\_inc_{(m,n),(i,j)} = \\
&OSIG0_{(m-1,n-1),(i,j)} + OSIG0_{(m-1,n),(i,j)} + OSIG0_{(m-1,n+1),(i,j)} + OSIG0_{(m+1,n-1),(i,j)} \\
&+ OSIG0_{(m,n),(i,j+1)} + OSIG0_{(m,n),(i+1,j)} + OSIG0_{(m,n),(i+1,j+1)} + OSIG0_{(m,n),(i',j')} \\
&+ OTF_{(m,n),(i-1,j-1)} + OTF_{(m,n),(i-1,j)} + OTF_{(m,n),(i,j-1)} + OTF_{(m,n),(i'',j'')}.
\end{aligned}
\tag{4.9}
$$

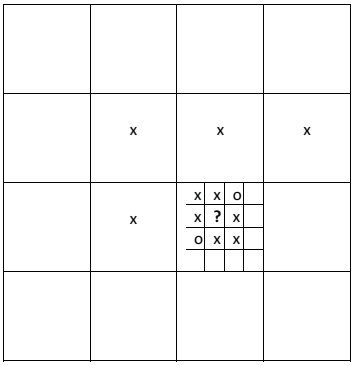

Figure 4.7: Context template for encoding $OSIG0$/ $OSIG1$/ $OSIG2$

In what follows, we discuss the case when $OBF_{(m,n)} = 0$, $m \in [1, H/8]$, $n \in [1, W/8]$.

[8]In this case, $OSIG0$s in (4.8) are substituted by $OSIG1$s or $OSIG2$s depending on which information is currently encoded.

In this case, only inliers are included in the block, and different statistics are observed, where the quantized indices are generally smaller compared to the case when $OBF_{(m,n)} = 1$. Therefore, a different coding scheme with different context models are designed for this case. A binary information called significant block flag (SBF) is encoded first to indicate if the block (m, n) has at least non-zero quantized index ($SBF_{(m,n)} = 1$) or not ($SBF_{(m,n)} = 1$), conditioned on already coded SBF and OBF information. The context template for SBF is illustrated in Fig. 4.8, where '?' is the current encoding SBF, 'x's denote the already encoded SBFs of adjacent blocks (if any), and 'OBF's represent the already encoded OBFs of its bottom and bottom-left neighbours. The corresponding context index increment is calculated by

$$\begin{aligned} SBF\_inc_{(m,n)} &= SBF_{(m-2,n)} + SBF_{(m-1,n-1)} \\ &+SBF_{(m-1,n)} + SBF_{(m-1,n+1)} + SBF_{(m,n-2)} \\ &+SBF_{(m,n-1)} + OBF_{(m,n+1)} + OBF_{(m+1,n)}. \end{aligned} \tag{4.10}$$

$SBF\_inc_{(m,n)}$ is truncated at 6, when the target distortion level $d < 34$ dB in PSNR. Otherwise, it is truncated at 7. So the number of context models reserved for SBF is up to 8. If $SBF_{(m,n)} = 1$ and $L_{(i,j)} > 0$, then a significant bit information $ISIG0$ is encoded to indicate if the corresponding index $c_{(m,n),(i,j)}$ is zero ($ISIG0_{(m,n),(i,j)}= 0$) or not ($ISIG0_{(m,n),(i,j)}= 1$). If $ISIG0_{(m,n),(i,j)}= 1$ and $L_{(i,j)} > 1$, a binary information $ISIG2$ is encoded to indicate if $c_{(m,n),(i,j)}$ is greater than 1 ($ISIG1_{(m,n),(i,j)}= 1$) or not ($ISIG1_{(m,n),(i,j)}= 0$). If $ISIG1_{(m,n),(i,j)}= 1$ and $L_{(i,j)} > 2$, another binary information $ISIG2$ is encoded to signal if $c_{(m,n),(i,j)}$ is greater than 2 ($ISIG2_{(m,n),(i,j)}= 1$) or not ($ISIG2_{(m,n),(i,j)}= 0$). $ISIG0$, $ISIG1$ and $ISIG2$ are encoded conditioned on their respective already coded information based on the context template shown in Fig. 4.5 . Their respective context index increments are calculated by (4.8) with OTFs replaced by $ISIG0$s, $ISIG1$s and $ISIG2$s accordingly, which are then truncated at 9, 7 and 7. The

contexts are finally merged based on Fig. 4.6, and those of $ISIG1$s and $ISIG2$s are merged together. So there are up to 70 context models reserved for encoding $ISIG0$, and up to 56 for $ISIG1$ and $ISIG2$. If $ISIG2_{(m,n),(i,j)}$= 1 and $L_{(i,j)} > 3$, $c_{(m,n),(i,j)} - 3$ is further coded, with the context index increments given in Fig. 4.6 for each frequency location.

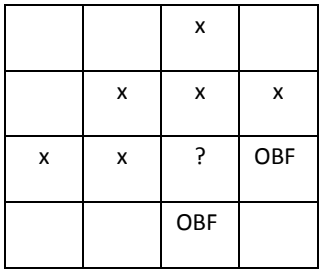

Figure 4.8: Context template for SBF suing already encoded OBFs of the 2 neighbours (bottom and bottom-left) and already encoded SBFs of its 6 neighbours

Lastly, we discuss how to encode the sign information when a quantized index is non-zero. In general, the correlations among the sign information are very weak. However, it is observed that there are some correlations among the co-located AC1s and AC2s (i.e., the two AC coefficients at the lowest frequency) across blocks. So the sign information is bypassed (1 bit is used) except for those of AC1 and AC2, which are denoted as $SGN1$ and $SGN2$, respectively. The context templates for encoding $SGN1$ and $SGN2$ are the same as OBF shown in Fig. 4.4. The context index increment of $SGN1$ for the block (m, n), $SGN1\_inc_{(m,n)}$, is given by

$$\begin{cases} max(SGN1\_inc'_{(m,n)}, -3) + 3, \text{ if } SGN1\_inc'_{(m,n)} < 0 \\ min(SGN1\_inc'_{(m,n)}, 3) + 3, \text{ otherwise} \end{cases}, \tag{4.11}$$

$$\text{where } SGN1\_inc'_{(m,n)} = SGN1_{(m-1,n-1)}$$
$$+SGN1_{(m-1,n)} + SGN1_{(m-1,n+1)} + SGN1_{(m,n-1)},$$

where $SGN1$=-1, 0, or 1 if the corresponding index is less than, equal to, or greater than 0. The corresponding context index increment of $SGN2$ is the same, except that $SGN1$s are replaced by $SGN2$s in (4.11). In total, there are up to 14 context models involved for encoding $SGN1$ and $SGN2$.

4.4 Soft-decision quantization design

Having developed the TCQ and the TCC, we now discuss how to design a RD optimization algorithm using the SDQ technique [79], [73], [74], [72], [70], [71], [16], [64], [22], in hopes of improving the RD performance of the proposed image coding system. In this thesis, we only consider SDQ for the inlier part, for the following two reasons: 1) the UQ in the TCQ for outliers are a purely uniform quantizers, given the reconstruction space, the HDQ outputs minimize the distortion, whereas, to find a better RD tradeoff, those of SDQ possibly sacrifice distortion to get gain in rate, which does not fit our design philosophy of this thesis, as we try to protect outliers from suffering too much of distortion; 2) the outliers are generally large in magnitudes and (can be seen as) uniformly distributed, so even if SDQ is considered, the gain in RD sense is very limited.

In this section, we propose a SDQ algorithm to jointly optimize the CDZQ and the CALCAC. The minimum coding unit in our case is a SL, and thus the proposed SDQ algorithm is scanning-layer-based, i.e., it operates SL wisely. So we call it (scanning) layer-based SDQ (LSDQ). In what follows, we formulate a optimization problem for LSDQ design in subsection 4.4.1, followed by a solution in subsection 4.4.2.

4.4.1 Layer-based SDQ design–Problem formulation

DCT is a unitary transform that maintains the Euclidean distances, which allows us to consider the optimization problem in DCT domain instead of image domain. Given an image in DCT domain $\mathbf{Y}$, our problem can be formulated as a constrained optimization over all possible index sequences $\mathbf{C}$:

$$\min_{\mathbf{C}} \|\mathbf{Y} - \hat{\mathbf{Y}}(\mathbf{C})\|^2, \text{ s.t. } R(\mathbf{C}) \leqslant R_{budget}, \tag{4.12}$$

or equivalently,

$$\min_{\mathbf{C}} R(\mathbf{C}), \text{ s.t. } \|\mathbf{Y} - \hat{\mathbf{Y}}(\mathbf{C})\|^2 \leqslant D_{budget}, \tag{4.13}$$

where $\|\mathbf{Y} - \hat{\mathbf{Y}}(\mathbf{C})\|^2$ is the distortion in DCT domain, $R(\mathbf{C})$ denotes the total compression rate resulting from encoding the index sequences $\mathbf{C}$. R_{budget} and D_{budget} are the rate constraint and distortion constraint, respectively.

4.4.2 Layer-based SDQ design–Problem solution

The original constrained optimization problem in (4.12) or (4.13) can be converted into an unconstrained one as follows by using Lagrange multiplier method

$$\min_{\mathbf{C}} J_{\theta'}(\mathbf{C}) = \|\mathbf{Y} - \hat{\mathbf{Y}}(\mathbf{C})\|^2 + \theta' \cdot R(\mathbf{C}), \tag{4.14}$$

where $J_{\theta'}(\mathbf{C})$ is Lagrangian cost or coding cost, and θ' is the Lagrange multiplier, which determines the relative tradeoff between rate and distortion. (4.14) is hard to solve, because of the inter-dependency of quantized indices among BNs and FNs. So we propose a sub-optimal solution to (4.16), i.e., the so-called LSDQ algorithm. Since the coding unit is a SL in the CALCAC, LSDQ is designed to minimize the coding cost for each SL, provided

the previously encoded index sequences are all fixed. Therefore, LSDQ is optimal within SLs.

The rate function in (4.14) can be estimated by empirical entropy, which is calculated using self-information. Let the quantized index at the sth scanning position in the current encoding SL be represented by c_s, and its absolute value denoted by $|c_s|$ ($|c_s| \leq L_s$), $s \in [1, N_{lyr}]$, where $N_{lyr} \in [1, 8]$ is the number of AC frequencies in the current SL. Given the corresponding context models '$\cdot$', the self-information of c_s, $I(c_s|\cdot)$, is calculated by

$$I(c_s|\cdot) = \begin{cases} -log(P(SIG0_s|\cdot)), \text{ if } |c_s| = 0 \\ -log(P(SIG0_s|\cdot)) + R(SGN_s|\cdot), \text{ if } |c_s| = 1 \text{ and } L_s = 1 \\ -log(P(SIG0_s|\cdot)) - log(P(SIG1_s|\cdot)) + R(SGN_s|\cdot), \text{ if } |c_s| = 1 \text{ and } L_s > 1, \text{ or } |c_s| > 1 \text{ and } L_s = 2 \\ -log(P(SIG0_s|\cdot)) - log(P(SIG1_s|\cdot)) - log(P(SIG2_s|\cdot)) + R(SGN_s|\cdot), \text{ if } |c_s| = 2 \text{ and } L_s > 2, \text{ or } |c_s| > 2 \text{ and } L_s = 3 \\ -log(P(SIG0_s|\cdot)) - log(P(SIG1_s|\cdot)) - log(P(SIG2_s|\cdot)) - log(P(|c|_s - 3|\cdot)) + R(SGN_s|\cdot), \text{ if } |c_s| > 2 \text{ and } L_s > 3 \end{cases} \tag{4.15}$$

where $SIG0/SIG1/SIG2$ can be $OSIG0/OSIG1/OSIG2$ or $ISIG0/ISIG1/ISIG2$ depending on the corresponding OBF value, log and P represent the base-2 Log function and the conditional probability of the encoding information based on its corresponding context models '$\cdot$', respectively, and $R(SGN_s|\cdot)$ is the rate for coding the sign information. Accordingly, the distortion resulting from compressing the AC y_s can be computed by

$$D(c_s) = \begin{cases} y_s^2, \text{ if } c_s = 0 \\ \| |y_s| - (\delta_s + q_s \cdot (|c_s| - 1)) \|^2, \text{ otherwise} \end{cases}.$$

Therefore, the corresponding Lagrangian cost is given by

$$J_{\theta'}(c_s|\cdot) = D(c_s) + \theta' \cdot I(c_s|\cdot). \tag{4.16}$$

Given the already coded information in the neighbouring SLs and neighbouring blocks, the coding cost of the current coefficient is only related to its immediate coded SLN (if any). In addition, since the coding cost resulting from compressing a AC as in (4.16) is additive, the total coding cost of a SL is simply the summation of that of every coefficient in the SL. As such, we propose a trellis structure to solve the optimization problem as shown in Fig. 4.9. In the trellis, each stage represents the quantization of a AC in the current SL. There are 4 states at each stage, denoting the quantized indices with magnitudes 0, 1, 2, or greater than 2. We then apply dynamic programming to efficiently solve the minimization problem.

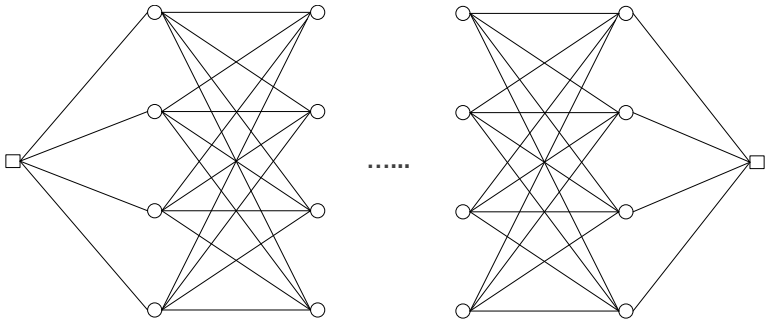

Figure 4.9: The trellis for the proposed LSDQ algorithm

Let $J_{\theta'}^{(s)}(t|t')$ denote the coding cost of the tth state at the sth stage of the trellis for the given $\theta' > 0$, with the incoming state from the $(s-1)$th stage being t', and $J_{\theta'}^{*(s)}(t)$ denote the corresponding optimal accumulating coding cost, $t, t' \in [1, 4]$, $s \in [1, N_{lyr}]$. LSDQ is summarized in Algorithm 6.

Algorithm 6 LSDQ algorithm for a SL (for a given θ').

1: Initialize $s = 1$, $J_{\theta'}^{*(0)}(t) = 0$ for $0 \leq t \leq 4$.

2: **while** $s \leq N_{lyr}$, **do**

3: **if** $t < 4$,

calculate $J_{\theta'}^{*(s)}(t) = \min_{1 \leq t' \leq 4} \left(J_{\theta'}^{*(s-1)}(t') + J_{\theta'}^{(s)}(t|t') \right)$, and label the corresponding optimal incoming state t'^{*};

4: **else** calculate $J_{\theta'}^{*(s)}(t) =$

$\min_{1 \leq t' \leq 4} \left(J_{\theta'}^{*(s-1)}(t') + \min_{2 < |\tilde{c}_s| \leq L_s} J_{\theta'}^{(s)}(t, \tilde{c}_s|t') \right)$, and label t'^{*} and the optimal index $\tilde{c}_s^{*}$ for the state $t = 4$;

5: **end if**

6: set $s = s + 1$;

7: **end while**

8: From the state t^{*} at the stage N_{lyr}, back track the optimal index sequence for the current SL until $s = 1$, according to the labels, where $t^{*} = \arg\min_{1 \leq t \leq 4} J_{\theta'}^{*(N_{lyr})}(t)$.

4.5 Multiresolution capability of the proposed image compression system

Having proposed the new image coding system and the relative RD optimization method, we now discuss its multiresolution capability in this section. Our system defines five multiresolution coding layers (MCLs) at the image level, i.e., MCL1-MCL5, in the syntax according to the data dependency. MCLs are defined in Table 4.1 in the encoding/decoding order.

Table 4.1: MCL definition in the proposed image compression system

MCL1	OBF info (i.e., OBFs of the whole image) in the CALBIC
MCL2	the rest of OTF info depending on the MCL1 in the CALBIC
MCL3	outlier info depending on the MCL1 & MCL2 in the CALCAC
MCL4	DC info, which is independent of other MCLs, in the CALCAC
MCL5	inlier info depending on the MCL1 & MCL2 in the CALCAC

At the encoder, the five MCLs are transmitted in sequence in the aforementioned order, i.e., the MCL1 is send to the decoder first, followed by the MCL2, if the channel condition allows, and then this scheme continues up to the MCL5, until the (channel coding) rate reaches the channel capacity. At the decoder, there are up to 5 reconstructed image versions with different resolution, as shown by an example in Fig. 4.10, where the compression rate of each version is listed in the caption.

As shown in Fig. 4.10, the MDI1 provides a general impression of the image, which is refined by the MDI2[9] with certain blurred edge and texture details, and further by

[9]Without the magnitude and sign information of outliers, the MDI2 is reconstructed based on the

the MDI3[10] offering the major skeleton of the image with some global unique edges and texture; higher resolution is obtained by the MDI4, providing the averaged texture in a blocky way while reserving sharper visual quality in some perceptually important areas, and the MDI5, yielding the fully decoded image. Among the low resolution MDIs, the MDI3, i.e., the decoded outlier image, is of the most interest, as it contains outliers, which convey some unique global image features in a clean description while taking relatively low compression rate. It can be used for image understanding applications such as image similarity analysis in image communications [76].

4.6 Experimental results

Since the proposed lossy image compression system is completely re-designed based on LPTCM, it is important to test coding efficiency of all the new designed components in a isolated manner at each coding unit level, as well as in a integrated manner at a system level. In this section, we demonstrate some experimental results of the coding/complexity performance of the proposed coding unit(s)/whole system (which are bolded in all tables). We have done five separate sets of tests as discussed in the following subsections.

4.6.1 Experimental results of the CDZQ

In the first test set, we intend to evaluate the theoretical compression performance of a CDZQ for truncated Laplacian (only nonnegative distributed part is considered in this

boundary parameter of the LPTCM for outliers, and inliers and DCs are set to 0, when performing the inverse DCT.

[10]Similar to the decoding of the MDI2, inliers and DCs are set to 0.

Figure 4.10: Multiresolution decoded images (MDI) for 512×512 Lena (when $d = 42$ dB in PSNR). Top-left: Original image; Top-middle: MDI1–up-sampled OBF image with a scale factor of 8 (Rate: 0.007 bpp), by MCL1 only; Top-right: MDI2–decoded OTF image (Rate: 0.081 bpp), by up to MCL2; Bottom-left: MDI3–decoded outlier image (Rate: 0.159 bpp), by up to MCL3; Bottom-middle: MDI4–decoded outlier image with DC (Rate: 0.704 bpp), by up to MCL4; Bottom-right: MDI5–fully decoded image (Rate: 1.294 bpp), by up to MCL5.

subsection), with the pdf given by

$$f(y) = \begin{cases} \frac{1}{2\lambda'(1-e^{-Y^c/\lambda'})} e^{-\frac{|y|}{\lambda'}} & \text{if } \mid y \mid < Y^c \\ 0 & \text{if } |y| > Y^c \end{cases}, \tag{4.17}$$

where Y^c is the truncation point in the truncated Laplacian distribution, and λ' denotes the scale parameter. Given Y and λ' and a distortion level d (that is achieved by the CDZQ), the CDZQ parameters L_{CDZQ}, i.e., the largest possible quantized index of the CDZQ (which solely decides the alphabet size, i.e., $L_{CDZQ}+1$) and q_{CDZQ}, i.e., the quantization step size of the CDZQ, can be determined via the table-looking procedure in Section 4.2. We calculate the rate by entropy in bit per symbol (bps) and distortion in PSNR based on truncated Laplacian distribution. Then by varying d, we have different RD points and thus obtain a RD curve, which represents the theoretical RD performance of a CDZQ for the given source. We intend to compare this RD curve with that of a DZQ [50], and the latter can be obtained by varying the quantization step size (of the DZQ). Another quantizers in comparison is ECSQ, where there is no constraints on the quantizers structures, i.e., no constraints for either quantization partitions or reconstruction levels. By varying the Lagrange multiplier, we can acquire a RD curve of the ECSQ for the given source. The number of reconstruction levels of the ECSQ is chosen to be the smallest one to achieve a specific RD point. Apart from the RD performance, we also intend to compare the number of reconstruction levels (needed to achieve those RD points), or the alphabet sizes, since they are important to entropy coding design–in general, the smaller the alphabet size is, the more efficient the entropy coding is.To this end, we seek for the RD points of the DZQ and ECSQ on the curves such that their respective achievable distortion are the same as the CDZQ, and record the corresponding alphabet sizes minus 1, i.e., L_{DZQ}s and L_{ECSQ}s, which are compared to L_{CDZQ}s, at the same distortion.

We have done simulations over different truncated Laplacian sources with various scale

parameters λ's and truncation points Y^cs. Due to the space limit, we select two representative source distributions and show the relative RD curves of the ECSQ, DZQ, and CDZQ in Fig. 4.11 and Fig. 4.12. The source with $\lambda' = 20$ and $Y^c = 120$ used to generate Fig. 4.11 is representative for a DCT source at high frequency, and that with $\lambda' = 4$ and $Y^c = 40$ for Fig. 4.12 is representative for a DCT source at low frequency. In the figures, L_{ECSQ}s, L_{DZQ}s and L_{CDZQ}s are shown corresponding to their respective RD points.

Our simulation results show that, on average, the ECSQ has the best theoretic compression performance for encoding truncated Laplacian sources, and that of the CDZQ is very close to the ECSQ and better than the DZQ, especially when encoding DCT sources at low frequency. For DCT sources at high frequency, the three quantizers perform (almost) equally good. On the other hand, the number of reconstruction levels needed by the CDZQ is generally smaller or equal to those of the ECSQ or DZQ, which will be helpful to subsequent entropy coding. This is because the constrain in (4.3) guarantee a small and efficient L_{CDZQ}. In addition, like the DZQ, the CDZQ maintains a very simple structure. As such, we adopt the CDZQ in this thesis rather than the ECSQ.

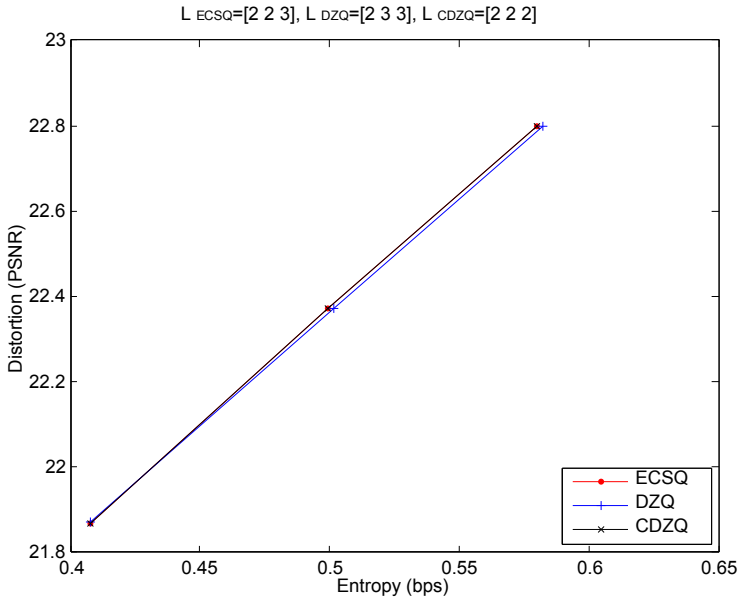

Figure 4.11: RD curve comparison for a DCT source at low frequency

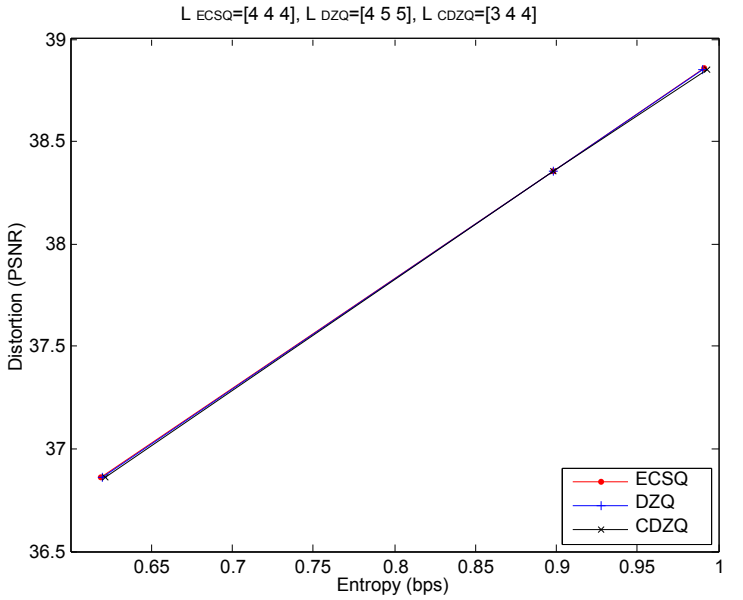

Figure 4.12: RD curve comparison for a DCT source at high frequency

4.6.2 Experimental results of the CALBIC

In this subsection, we show some experimental results of the CALBIC in (losslessly) encoding bi-level images of OTFs. On average, the OTF image generally consumes less than 10% of overall rate. Table 4.2 and Table 4.3 compare the compression rate (in total bits) of the proposed CALBIC and the state-of-the-art bi-level image coding method JBIG2 [3], where the test images are grey-scale 512×512 'Lena' and 512×512 'GoldHill', respectively. In the tables, ds represent the target distortion levels. Different d will produce different bi-level images according to the OptD technique, with $d = 30$ dB and $d = 42$ dB in PSNR indicating low-rate and high-rate encoding, respectively. The gain on rate savings of the CALBIC over JBIG2 is shown in the last column.

On average, the CALBIC reduces the compression rate roughly by 1/4, compared to JBIG2. The reason why the gain is such significant is that, the CALBIC is able to better capture data dependencies inside one block and at the same frequency location across adjacent blocks. The gain tends to be larger in low-rate coding (when d is small in PSNR). This is because that JBIG2 has a large context template, and it tends to perform poorly when rate is low in encoding the OTF images.

Table 4.2: Rate comparison for OTF image of 512×512 Lena

d (in dB)	JBIG2 (in bits)	CALBIC (in bits)	gain
30	20616	**14739**	-29%
42	26432	**21280**	-20%

Table 4.3: Rate comparison for OTF image of 512×512 GoldHill

d (in dB)	JBIG2 (in bits)	CALBIC (in bits)	gain
30	8740	**6797**	-25%
42	14272	**11444**	-20%

4.6.3 Experimental results of the TCC

Having tested the coding performance of the CALBIC, we now evaluate the TCC (the CALBIC + the CALCAC) as a whole lossless coding system. In this subsection, the TCC is applied to encode the residuals extracted from HEVC codec (HM 10.1) [32]. Specifically, HEVC lossless mode (i.e., transform-quantization bypass mode) is enabled. Our coding procedure is described as follows. First, acquire the residual image after the HEVC intra prediction for the whole image is done. Then, apply the invertible integer DCT proposed in [38] to the residual image on each 8×8 block. Lastly, perform the TCC to encode the integer DCT coefficients. The compression rate consists of two part: 1) HEVC intra prediction overhead mainly including coding unit (CU) partition and prediction mode information, etc.; 2) bit consumption from the TCC. Table 4.4 and Table 4.5 show the lossless coding rate in bit per pixel (bpp) of the TCC for encoding grey-scale 512×512 'Lena' and 512×512 'GoldHill', respectively, which is compared to HEVC lossless mode (HEVC-LS). It is also compared to those of some popular lossless image coding methods, such as JPEG-LS [4], JPEG 2000 lossless (JPEG2000-LS) [5], and CALIC [63].

It is shown that our tested lossless coding scheme using the TCC outperforms HEVC-LS with an average 7% bit savings. In general, it is slightly better than JPEG-LS and JPEG2000-LS, with a small performance loss compared to CALIC. Note that CALIC (as well as JPEG-LS and JPEG2000-LS) is not designed for DCT-based lossy image coding, so

data dependencies among DCT coefficients can not be fully exploited. This result in very poor coding performance when encoding quantized DCT coefficients output by the TCQ, which is verified by our experiments. In this case, for fair comparison, we will compare our proposed image coding system as a whole with other lossy image coding methods in the next subsection.

Table 4.4: Lossless coding rate comparison for 512×512 Lena

HEVC-LS	JPEG-LS	JPEG2000-LS	CALIC	TCC
4.497	4.245	4.316	4.095	**4.190**

Table 4.5: Lossless coding rate comparison for 512×512 GoldHill

HEVC-LS	JPEG-LS	JPEG2000-LS	CALIC	TCC
5.107	4.712	4.837	4.604	**4.727**

4.6.4 Experimental results of the proposed image coding system– subjective tests

In this subsection, we move on to our fourth test set to evaluate the coding performance of the whole lossy coding system in terms of compression rate vs subjective quality. The decoded images of our proposed image coding are visually compared with those of the state-of-the-art DCT-based non-predictive image coding method ECEB proposed in [56] and baseline JPEG [1], and those of some DCT-based predictive image coding methods including H.264/AVC (JM 18.3 High Profile) [31] and HEVC (HM 10.1) [32] intra coding. Only HDQ cases are tested. For fair comparison, the transform sizes in both H.264/AVC and HEVC have been set to 8×8, and all post-processing filters are disabled. Experimental

results show that it is very difficult to observe differences among the decoded images of all test coding schemes at high-rate end. However, the image quality differs a lot for low-rate cases. Due to the space limit, we only show a representative example of the decoded images of (part of) 512×512 'Lena' at low-rate in Fig. 4.13, and similar observation (as discussed later) can be seen in other tested images. As a quantitive measure to the subjective quality, SSIM [59] tests have been done, with the relative results shown in the caption of the figure. Also listed are PSNR results for each decoded image. The rate for all tested coding schemes has been adjusted to the same level around 0.26 bpp.

By examining the figure (zoom in if possible, as the sub-figures are small), one can see that the decoded image of the proposed is sharper and has much less blocking effects, compared to those of its non-predictive coding counterparts, i.e., JPEG and the ECEB. It also has a significant better SSIM score. This is because the protection of outliers keeps higher fidelity of some visually important areas. Interestingly, although the decoded images of the ECEB and the proposed have the similar PSNR values (indicating similar objective quality), the latter yields much better visual quality. Exactly the same observation can be found in the low-rate coding of some other images, such as 512×512 Barbara.

Despite the PSNR gap between the proposed and the predictive coding, the subjective quality of their decoded images is at the same level. They also have similar SSIM scores, with the proposed marginally better than H.264/AVC and slightly worse than HEVC. If examine carefully, to some extent, the decoded image of the proposed is even visually closer to the original image, and looks more agreeable. For instant, the decoded image of H.264/AVC renders lower visual quality at some perceptually important areas, such as the bottom eyelash of Lena's right eye. In addition, even though it looks smoother with less blocking effects than the proposed, the decoded image of HEVC has very unpleasant ringing artifacts resulting from the prediction, which appear in many visually observable

Figure 4.13: Subjective image quality comparison for 512×512 Lena coded at 0.26 bpp. Top-left: original image (SSIM: 1, PSNR: inf dB); Top-middle: ECEB (SSIM: 0.942, PSNR: 33.32 dB); Top-right: the proposed (SSIM: 0.953, PSNR: 33.39 dB); Bottom-left: JPEG (SSIM: 0.931, PSNR: 31.90 dB); Bottom-middle: H.264/AVC (SSIM: 0.952, PSNR: 34.10 dB); Bottom-right: HEVC (SSIM: 0.959, PSNR: 34.84 dB)

places, such as the bottom eyelash of Lena's right eye, the up-left of her lip, and edge of the board (in the background) that closes to her hair, etc.

4.6.5 Experimental results of the proposed image coding system– objective tests

Lastly, as the focus of this section, we evaluate the coding performance of the proposed DCT-based non-predictive lossy coding system in terms of compression rate vs objective quality, as well as complexity expense. Note that the proposed is a non-predictive image coding system, in the sense that there is no image-domain prediction (such as the intra prediction used in H.264/AVC or HEVC) involved.

Since we target at real-world image compression application, to decide our rate of interest in the experiments, we have investigated the rate range yielded by some digital cameras in compressing raw images using baseline JPEG. In practice, the JPEG encoders on these devices apply one or more predefined quantization tables based on target (decoded) image quality, or target distortion equivalently. Table 4.6 shows the only quantization table for luminance employed by the JPEG encoder on the popular Apple iPhone 4/4S models. Since the quantization step sizes are generally very small, especially for those of low frequencies that contain most of the energy of the encoded image, the compression rate shall be very high. As we observed, the rate in bpp is in the range of 2.5 to 4.0. Since we use the OptD technique in the proposed system, we desire to convert this rate range to the target distortion (i.e., d) range. As a result, this rate range usually corresponds to the d ranged from 42 dB to 50 dB in PSNR in our experiments. On the other hand, low-rate image coding is still of interest for research purposes. So we decide the range of d to be from 30 dB to 50 dB with a step size of 2 dB in the fifth test set (to show the coding

performance with a wide range of rate). However, in practice, there is no need to aim at so many fidelity levels, 3 or 4 is enough in general depending on different applications.

Table 4.6: The predefined quantization table for the luminance used for baseline JPEG encoding on an Apple iPhone 4/4S

1	1	1	1	2	5	6	5
1	1	1	2	4	5	6	5
1	1	2	2	5	8	8	6
1	2	3	5	6	10	10	7
2	3	5	6	8	10	11	9
4	6	7	8	10	12	12	10
7	9	9	9	11	10	10	9
4	6	6	7	8	10	9	9

Experiments have been done to evaluate the proposed image compression system on a set of standard 8-bit grayscale test images with various sizes. Table 4.7–Table 4.11 show the PSNR performance of the proposed system for 512×512 'Lena', 512×512 'Barbara', 1024×1024 'Airfield', 720p 'Parkrun' (the 1st frame), and 1080p 'BQTerrace' (the 1st frame), with its HDQ (i.e., SDQ is off) and SDQ versions referred to as the Prop. HDQ and the Prop. SDQ, respectively. The first column in each table denotes the resulting compression rate of the Prop. HDQ[11] in bpp when the distortion level d is set in the aforementioned range. The compression performance is compared with the state-of-the-art non-predictive image coding ECEB[12]. The uniform quantization table used in the ECEB code is then optimized by the OptD technique according to the algorithm proposed in [69]. This coding scheme is dubbed ECEB-OptD, with its related PSNR results illustrated in the tables. The coding performance is also compared with that of some popular non-

[11] The rate for other coding methods has been adjusted to the same rate as that of the Prop. HDQ, either by adapting their respective coding parameters (for all non-predictive coding methods) or by interpolation (for all predictive coding methods).

[12] All the ECEB related tests are run on the code kindly provided by the authors of [56].

predictive/predictive image/video coding standards, including JPEG, H.264/AVC intra coding (with RDOQ off/on referred to as H.264/AVC HDQ/H.264/AVC RDOQ, respectively), and the newly developed HEVC intra coding (with RDOQ off/on referred to as HEVC HDQ/HEVC RDOQ, respectively). The softwares used for H.264/AVC and HEVC intra coding are JM 18.3 High Profile and HM 10.1, respectively. For fair comparison, the transform sizes in both H.264/AVC and HEVC have been set to 8×8. Table 4.12 - Table 4.14 provide experimental results of overall computational complexity, showing by CPU running time in milliseconds[13] for 512×512 'Lena', 1024×1024 'Airfield', and 720p 'Parkrun', respectively.

As shown in the tables, the proposed image coding significantly outperforms baseline JPEG by more than 4.3 dB in PSNR on average with some complexity increase, which is still at a reasonable level for real-time image communications. Compared with the ECEB/ECEB-OptD, the the Prop. HDQ and the Prop. SDQ provide an average 0.75/0.4 dB and 1/0.65 dB performance gain in PSNR, respectively. the Prop. HDQ outperforms both the ECEB and ECEB-OptD for all the rate points and for all the test images. The gain tends to be larger when the rate is higher, and it is limited when the rate is low. The reasons contributed to this are 1) the overhead of the proposed image coding takes up a non-negligible rate consumption in low-rate case, which compensates some performance gain; 2) the room to improve the coding efficiency is limited in low-rate case (for non-predictive image coding). According to our experiments, CALIC type of DC (quantized coefficient domain) prediction as in [63] saves about 5% of bit rate (in coding DC alone) compared to the simple predictive coding for DC in the ECEB. But when average out for 64 DCT coefficients, this gain is negligible. So the gain over the ECEB mainly comes from the OptD and the combination of the TCQ and the TCC, and that over the ECEB-OptD

[13] All the simulations are run on an Apple Mac Pro 8-core 2.4GHz 12G RAM workstation.

mainly from the latter, which actually benefited from our protection of outliers in the TCQ and special coding schemes of outliers in the TCC. As for complexity comparison, the proposed provides the overall computational complexity that is at the same level of the ECEB/ECEB-OptD, with negligible increase of storage complexity for the HDQ case, and some complexity increase for the SDQ case. It has generally lower computational complexity when the rate is low, thanks to the OptD technique and the multi-symbol arithmetic coding.

In comparison to the predictive image coding methods, the proposed provides an overall 0.4 dB gain or so over H.264/AVC, with dramatically reduced computational complexity. Compared to the state-of-the-art HEVC intra coding, the proposed provides comparable or sometimes even better compression performance at the high-rate end, or when encoding some complicated images (such as the 720p 'Parkrun') where the intra prediction is not that efficient, with more than 95% of the computational complexity saved. However, as one can see, the proposed image coding still has a significant performance gap compared to either H.264/AVC or HEVC in the low-rate region, due to the lack of intra prediction. It is hard, if not impossible, to apply intra prediction to our proposed image coding system based on the LPTCM, since it is coupled with quantization. Whereas, the proposed TCQ requires to know the LPTCM distribution of the residual image, which is not available until the intra prediction for the whole image is done.

4.7 Chapter review

Motivated by the special characteristics of outliers of the LPTCM, in this chapter, we aim at exploring its applications to image compression and investigating possible improvement on both subjective and objective image quality. To this end, we have proposed an effi-

Table 4.7: PSNR performance comparison of different coding methods for 512x512 Lena

Rate	JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
0.264	31.90	33.32	33.35	**33.39**	**33.51**	34.54	34.70	35.29	35.44
0.338	33.11	34.49	34.45	**34.61**	**34.77**	35.56	35.73	36.27	36.40
0.441	34.34	35.87	35.76	**35.94**	**36.16**	36.70	36.85	37.31	37.51
0.584	35.54	37.23	37.21	**37.43**	**37.62**	37.83	38.05	38.42	38.63
0.792	36.90	38.69	38.81	**38.99**	**39.18**	39.15	39.41	39.67	39.96
0.968	37.75	39.69	39.86	**40.03**	**40.23**	40.06	40.35	40.63	40.90
1.294	39.19	41.33	41.59	**41.77**	**41.96**	41.66	41.93	42.25	42.50
1.660	40.45	42.99	43.39	**43.64**	**43.81**	43.39	43.65	44.14	44.34
2.030	41.68	44.83	45.27	**45.58**	**45.71**	45.32	45.48	46.12	46.32
2.386	42.80	46.44	47.09	**47.50**	**47.64**	47.16	47.35	48.10	48.28
2.746	44.03	47.95	48.97	**49.42**	**49.57**	48.98	49.18	49.93	50.14
3.086	45.30	49.18	50.63	**51.13**	**51.28**	50.62	50.84	51.59	51.87

Table 4.8: PSNR performance comparison of different coding methods for 512x512 Barbara

Rate	JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
0.545	28.96	32.37	32.29	**32.39**	**32.54**	32.84	33.02	33.75	34.00
0.683	30.45	33.95	33.80	**34.00**	**34.20**	34.33	34.55	35.22	35.44
0.840	31.95	35.50	35.28	**35.61**	**35.82**	35.82	36.01	36.63	36.90
1.042	33.66	37.20	37.02	**37.38**	**37.60**	37.37	37.64	38.18	38.44
1.285	35.38	38.89	38.85	**39.14**	**39.39**	38.98	39.27	39.75	40.07
1.458	36.45	39.94	40.00	**40.25**	**40.52**	39.99	40.30	40.73	41.09
1.760	38.09	41.57	41.81	**42.03**	**42.25**	41.57	41.92	42.34	42.69
2.096	39.72	43.26	43.58	**43.84**	**44.05**	43.21	43.54	44.06	44.40
2.453	41.35	44.95	45.23	**45.73**	**45.91**	45.02	45.24	45.93	46.21
2.810	42.71	46.54	46.99	**47.59**	**47.64**	46.76	46.95	47.83	48.08
3.164	44.07	48.00	48.63	**49.40**	**49.58**	48.33	48.47	49.62	49.89
3.516	45.53	49.26	50.14	**51.15**	**51.31**	49.61	49.79	51.35	51.67

Table 4.9: PSNR performance comparison of different coding methods for 1024x1024 Airfield

Rate	JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
0.599	29.92	31.03	31.13	**31.28**	**31.41**	31.32	31.48	31.62	31.81
0.858	30.97	32.59	32.70	**32.83**	**32.97**	32.80	33.00	33.12	33.32
1.192	32.14	34.36	34.47	**34.59**	**34.79**	34.50	34.73	34.91	35.13
1.558	33.40	36.25	36.34	**36.52**	**36.73**	36.38	36.55	36.81	37.02
1.929	34.59	38.22	38.36	**38.52**	**38.76**	38.30	38.42	38.80	38.98
2.145	35.33	39.38	39.53	**39.74**	**40.00**	39.46	39.55	40.00	40.18
2.504	36.59	41.32	41.53	**41.78**	**42.05**	41.42	41.52	42.02	42.21
2.852	37.92	43.14	43.43	**43.80**	**44.06**	43.35	43.46	43.96	44.20
3.196	39.35	44.87	45.22	**45.75**	**45.99**	45.26	45.41	45.86	46.15
3.539	40.88	46.48	47.01	**47.64**	**47.88**	47.12	47.23	47.73	48.06
3.888	42.54	47.95	48.87	**49.49**	**49.73**	48.76	48.85	49.53	49.90
4.229	44.42	49.20	50.45	**51.18**	**51.42**	49.91	50.02	51.24	51.63

Table 4.10: PSNR performance comparison of different coding methods for 720p Parkrun (1st video frame)

Rate	JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
1.093	28.43	31.25	31.25	**31.29**	**31.54**	31.48	31.70	31.50	31.74
1.355	29.78	33.03	32.99	**33.11**	**33.35**	33.17	33.42	33.23	33.51
1.642	31.20	34.80	34.79	**34.94**	**35.17**	34.89	35.15	35.04	35.34
1.953	32.69	36.63	36.69	**36.81**	**37.06**	36.70	36.94	36.88	37.21
2.279	34.24	38.48	38.61	**38.76**	**38.99**	38.53	38.76	38.75	39.13
2.483	35.24	39.62	39.78	**39.96**	**40.22**	39.65	39.86	39.88	40.27
2.827	36.62	41.48	41.69	**41.94**	**42.18**	41.50	41.69	41.78	42.17
3.167	38.23	43.27	43.54	**43.87**	**44.08**	43.06	43.12	43.66	44.04
3.512	39.77	45.01	45.46	**45.80**	**46.02**	44.39	44.47	45.59	45.96
3.859	41.12	46.61	47.31	**47.67**	**47.88**	45.66	45.75	47.49	47.91
4.203	42.55	48.05	49.14	**49.49**	**49.69**	46.78	46.91	49.35	49.81
4.550	44.30	49.30	50.79	**51.21**	**51.42**	47.86	48.01	51.19	51.68

Table 4.11: PSNR performance comparison of different coding methods for 1280p BQTerrace (1st video frame)

Rate	JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
0.432	30.18	32.08	32.04	**32.20**	**32.36**	32.81	32.95	33.94	34.15
0.578	31.66	33.81	33.68	**33.90**	**34.12**	34.47	34.66	35.49	35.75
0.744	32.98	35.30	35.34	**35.45**	**35.64**	35.98	36.22	36.97	37.36
0.961	34.38	37.19	37.09	**37.36**	**37.66**	37.68	38.07	38.80	39.22
1.255	36.11	39.75	39.58	**40.01**	**40.35**	39.94	40.42	41.30	41.66
1.439	36.98	41.41	41.35	**41.64**	**41.89**	41.47	41.93	42.85	43.19
1.681	38.16	43.53	43.41	**43.92**	**44.24**	43.48	43.90	44.84	45.15
1.867	39.04	45.03	44.87	**45.66**	**45.92**	44.94	45.28	46.23	46.54
2.068	40.44	46.30	46.43	**47.18**	**47.43**	46.45	46.70	47.63	47.95
2.293	42.16	47.55	48.06	**48.86**	**49.13**	47.86	48.15	49.21	49.48
2.532	44.81	48.75	49.54	**50.60**	**50.88**	49.38	49.65	50.67	50.99
2.813	47.05	49.86	51.32	**52.30**	**52.58**	50.83	51.07	52.21	52.52

Table 4.12: Computer running time (in milliseconds) of different image coding methods for encoding 512×512 Lena

JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
20	51	52	**63**	**148**	484	729	1433	1927

Table 4.13: Computer running time (in milliseconds) of different image coding methods for encoding 1024×1024 Airfield

JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
83	336	337	**396**	**878**	2560	4814	6607	9058

Table 4.14: Computer running time (in milliseconds) of different image coding methods for encoding 720p Parkrun (1st video frame)

JPEG	ECEB	ECEB-OptD	Prop. HDQ	Prop. SDQ	H.264 HDQ	H.264 RDOQ	HEVC HDQ	HEVC RDOQ
71	294	295	**357**	**813**	2382	3730	9063	11817

cient DCT-based non-predictive image coding system, where outliers are protected from suffering too much distortion considering its subjective and objective importance, and handled separately from inliers considering their statistical difference. A bi-level image coding method called CALBIC is first developed to encode the outlier flags. With them as side information, the so-called TCQ is then proposed to quantize the outlier by a UQ and the inlier by the CDZQ, followed by the composite entropy coding method CALCAC to encode the quantized indices of the outlier and inlier separately. Lastly, a SDQ algorithm is presented to jointly optimize the CDZQ and CALCAC to further improve the RD performance of the proposed system. To evaluate our proposed coding unit(s)/whole system, we have done five separate sets of experiments as follows. In the first set, to test its compression performance, we apply the CDZQ on random sources and calculate the resulting rate by the entropy and distortion by MSE. The theoretical RD performance is then compared with that of a DZQ, showing that the former is generally superior than the latter. The second one is to test the compression rate of the CALBIC in (losslessly) encoding bi-level images of outlier flags. Compared to JBIG2 [3], the state-of-the-art bi-level image coding, remarkably less amount of bits are needed by the CALBIC. The purpose of our third test set is to evaluate the lossless coding efficiency of the TCC, i.e., CALBIC + CALCAC.

Compared with HEVC entropy coding in lossless mode (with the same coding settings), the TCC provides significantly better coding performance. Next, the fourth test set is to evaluate the coding performance of the lossy coding system as a whole in terms of compression rate vs subjective quality (of decoded images). It is shown that our proposed system achieves compression results that are among the best and similar to those of H.264/AVC or HEVC intra (predictive) coding. Lastly, in the fifth test set, the compression performance of the proposed system in terms of coding rate vs objective quality is tested, as well as complexity expense. When applied to encode a set of standard test images, our proposed image coding significantly outperforms any non-predictive coding including the state-of-the-art ECEB [56], and even some predictive coding including H.264/AVC intra coding [31], with relatively low computational complexity that is at the same level as the ECEB, and negligible increase of storage complexity. In comparison to the newly developed HEVC [32] intra coding, the proposed system provides comparable or sometimes even better compression performance at the high-rate end, or when encoding some complicated images, with dramatically reduced computational complexity. This, together with its ability to improve subjective image quality, makes our proposed image compression system a perfect compromise between coding efficiency and complexity expense.

An interesting byproduct of the proposed system is its multiresolution capability. Thanks to the CALBIC scheme and the separate coding scheme of outliers and inliers in the CALCAC, it is possible to yield several decoded versions from the bitstream depending on the transmission channel capacity. This appealing feature, together with the comparatively high coding efficiency and low complexity, makes it a good alternative for real-time image processing applications.

Chapter 5

Conclusion and future work

5.1 Conclusion

To push the envelope of DCT-based lossy image/video compression, this thesis has proposed some novel perspectives from source modelling, to quantization table design, to quantizers and entropy coding design. To better handle the heavy tail phenomenon commonly seen in DCT coefficients, a new model called TCM has been first developed and justified. Given a sequence of DCT coefficients, the TCM first separates the tail from the main body of the sequence, and then uses a uniform distribution to model DCT coefficients in the heavy tail, while using a parametric distribution to model DCT coefficients in the main body. The separation boundary and other distribution parameters are estimated online via ML estimation. Efficient online algorithms are proposed for parameter estimation and their convergence is also proved. When the parametric distribution is truncated Laplacian, the resulting TCM dubbed LPTCM not only achieves superior modelling accuracy with low estimation complexity, but also has a good capability of nonlinear data reduction by separating outliers from inliers.

Quantization table design has then been revisited for image/video coding where soft decision quantization (SDQ) is considered. Unlike conventional approaches where quantization table design is bundled with a specific encoding method, we assume optimal SDQ encoding and design a quantization table for the purpose of reconstruction. Under this assumption, we model transform coefficients across different frequencies as independently distributed random sources and apply the Shannon lower bound to approximate the rate distortion function of each source. We then show that a quantization table can be optimized in a way that the resulting distortion complies with certain behavior, yielding the so-called optimal distortion profile scheme (OptD). Guided by this new theoretical result, we present an efficient statistical-model-based algorithm using the Laplacian model to design quantization tables for DCT-based image compression. When applied to standard JPEG encoding, it provides significant performance gain (in PSNR), with almost no extra burden on complexity. The proposed algorithm notably outperforms the state-of-the-art JPEG quantization table optimizer, with huge computational complexity reduced. Significant compression performance improvement is also seen when it is applied to other image coding systems proposed in the literature.

Lastly, based on the LPTCM and OptD, we further propose an efficient non-predictive DCT-based image compression system, where outliers are protected from suffering too much distortion considering its subjective and objective importance, and handled separately from inliers considering their statistical difference. A bi-level image coding method called CALBIC is first developed to encode the outlier flags. With them as side information, the so-called TCQ is then proposed to quantize the outlier by a UQ and the inlier by the CDZQ, followed by the composite entropy coding method CALCAC to encode the quantized indices of the outlier and inlier separately. Lastly, a SDQ algorithm is presented to jointly optimize the CDZQ and CALCAC to further improve the RD performance of

the proposed system. It is shown that our new system achieves the best non-predictive image coding performance in terms of both rate vs visual quality and rate vs PSNR, with very low complexity. Compared to the state-of-the-art predictive image coding, i.e., HEVC intra coding, it provides similar coding performance in terms of rate vs visual quality, and comparable or even better coding performance in terms of rate vs PSNR in the high-rate region or for some complicated images, with huge computational complexity reduced. In addition, the proposed image coding system also provides a multiresolution feature, which, together with its comparatively high coding efficiency and low complexity, makes it a good alternative for real-time image processing applications.

5.2 Future work

There are many topics left for future work. In the following, we discuss a few of them.

1) As discussed in Section 4.6, it is hard to apply intra prediction to our proposed image coding, since the prediction and quantization is bundled together. However, given an inter-predicted residual image of the previous video frame, it might be possible to use our new image compression method based on the LPTCM or GMTCM to improve inter video coding performance, such as HEVC inter coding, which will be one of our future work.

2) Another interesting future work is to use the LPTCM or GMTCM to improve compression/complexity performance for the in-loop filters in video coding, e.g., HEVC inter coding. In HEVC, in-loop filtering is performed after the prediction for the whole frame is done, and at that time the residue of the whole frame is available for the LPTCM or GMTCM to apply. There are two types of in-loop filters in HEVC–deblocking filters and sample adaptive offset (SAO) filters [32]. For deblocking filters, natural edges should not

be filtered [27], [34]. In the HEVC codec, edges are calculated by gradients pixel by pixel on the largest coding unit (LCU) boundaries [32], which is time-consuming and may not be capable of detecting all the edges. To solve these issues, the (quantized) outlier image that is available at both the encoder and decoder (based on the first future work) can be incorporated into the deblocking process, which contains edges and other global features that are perceptually important and are better not be filtered. For SAO filters, the outlier image can be used to find samples needed to be offset (such that smaller distortion is achieved) more efficiently, so that time complexity can be reduced.

Bibliography

[1] ISO/IEC 10918-1 and ITU-T Recommendation T.81. Information technology - digital compression and coding of continuous-tone still images: Requirements and guidelines, 1994.

[2] ISO/IEC 11544 and ITU-T Recommendation T.82. Information technology - coded representation of picture and audio information – progressive bi-level image compression, 1993.

[3] ISO/IEC 14492 and ITU-T Recommendation T.88. Information technology - lossy/lossless coding of bi-level images, 2001.

[4] ISO/IEC 14495 and ITU-T Recommendation T.87. Information technology lossless and near-lossless compression of continuous-tone still images baseline, 1998.

[5] ISO/IEC 15444-1 and ITU-T Recommendation T.800. Information technology - JPEG2000 image coding system: Core coding system, 2000.

[6] T. Acharya and A. K. Ray. *Image Processing - Principles and Applications*. Wiley InterScience, 2006.

[7] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23:90–93, January 1974.

[8] G. Bjntegaard and K. Lillevold. Context-adaptive VLC coding of coefficients. In *Proceedings of the JVT-C028, Joint Video Team (JVT) ISO/IEC MPEG, ITU-T VCEG, 3rd Meeting*, pages 6–10, May 2002.

[9] F. Bossen, B. Bross, K. Suhring, and D. Flynn. Hevc complexity and implementation analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1685–1696, December 2012.

[10] W.-H. Chen, C. Smith, and S. Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transaction on Communications*, 25:1004–1009, September 1997.

[11] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017, available online at: http://www.cisco.com/en/us/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white paper c11-520862.html.

[12] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley & Sons,, New York, 2006.

[13] M. Crouse and K. Ramchandran. Joint thresholding and quantizer selection for decoder-compatible baseline jpeg. In *Proceedings of the 1995 IEEE Acoustics, Speech, and Signal Processing, (ICASSP '95)*, pages 2331–2334, September 1995.

[14] M. Crouse and K. Ramchandran. Joint thresholding and quantizer selection for transform image coding: Entropy constrained analysis and applications to baseline JPEG. *IEEE Transaction on Image Processing*, 6:285–297, February 1997.

[15] M. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and Kullback-Leibler distance. *IEEE Transaction on Image Processing*, 11:146–158, February 2002.

[16] E.-H.Yang and C. Sun. Dithered soft decision quantization for baseline JPEG encoding and its joint optimization with Huffman coding and quantization table selection. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, pages 249–253, November 2011.

[17] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer, Norwell, MA, 1992.

[18] R.M. Gray and T.G. Stockham Jr. Dithered quantizers. *IEEE Transaction on Information Theory*, 39:805–812, May 1993.

[19] J. Huang and T. Meng. Optimal quantizer step sizes for transform coders. In *Proceedings of the IEEE Acoustics, Speech, and Signal Processing*, pages 2621–2624, April 1991.

[20] International Telecommunication Union (ITU). Recommendation h.263: video coding for low bit-rate communications, 1998.

[21] N. Kamaci, Y. Altunbasak, and R. M. Mersereau. Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(8):994–1006, 2005.

[22] M. Karczewicz, P.S. Chen, Y. Ye, and R. Joshi. R-d based quantization in h.264. In *Proceedings of the SPIE Applications of Digital Image Processing*, volume 7443, pages 744314–744314–8, September 2009.

[23] J. Kieffer. A survey of the theory of source coding with a fidelity criterion. *IEEE Transactions on Information theory*, 39:1473 – 1490, September 1993.

[24] J. C. Kieffer and E.-H. Yang. Grammar based codes: A new class of universal lossless source codes. *IEEE Transactions on Information Theory*, 46:737–754, May 2000.

[25] E. Y. Lam and J. W. Goodman. A mathematical analysis of the DCT coefficient distributions for images. *IEEE Transaction on Image Processing*, 9:1661–1666, October 2000.

[26] T. Linder and R. Zamir. On the asymptotic tightness of the shannon lower bound. *IEEE Transaction on Information Theory*, 40:2026–2031, November 1994.

[27] P. List, A. Joch, J. Lainema, G. Bjntegaard, and M. Karczewicz. Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):614–619, July 2003.

[28] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:620–636, July 2003.

[29] N. Memon. Adaptive coding of dct coefficients by Golomb- Rice codes. In *Proceedings of the IEEE International Conference on Image Processing*, pages 516–520, Octobor 1998.

[30] ISO/IEC MPEG. Coding of audio-visual objects-part 2: Visual, ISO/IEC 14 496-2 (MPEG-4 Visual Version 1), 1999.

[31] ISO/IEC MPEG and ITU-T VCEG. ITU-T recommendation h.264: Advanced video coding for generic audiovisual services, 2005.

[32] ISO/IEC MPEG and ITU-T VCEG. High efficiency video coding (HEVC) text specification draft 10, 2013.

[33] F. Muller. Distribution shape of two-dimensional DCT coefficients of natural images. *Electronics Letters*, 29:1935–1936, October 1993.

[34] A. Norkin, G. Bjntegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. H. Zhou, and G. V. d. Auwera. Hevc deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1746–1754, December 2012.

[35] A. Ortega and K. Ramchandran. Rate-distortion menthods for image and video compression. *IEEE Signal Processing Magazine*, 5:23–50, November 1998.

[36] I.-M. Pao and M. T. Sun. Modeling DCT coefficients for fast video encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:608–616, June 1999.

[37] W. B. Pennebaker and J. L. Mitchell. JPEG*: Still Image Data Compression Standard*. Kluwer, Norwell, Massachusetts, 2003.

[38] G. Plonka and M. Tasche. Invertible integer DCT algorithms. *Applied and Computational Harmonic Analysis*, 15:70–88, July 2003.

[39] K. Ramchandran and M. Vetterli. Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility. *IEEE Transaction on Image Processing*, 3:700–704, September 1994.

[40] V. Ratnakar, E. Feig, E. Viscito, and S. Kalluri. Runlength encoding of quantized discrete cosine transform (DCT) coefficients. In *Proceedings of the SPIE*, volume 2419, pages 398–406, April 1995.

[41] V. Ratnakar and M. Livny. RD-OPT: an efficient algorithm for optimizing DCT quantization tables. In *Proceedings of the Data Compression Conference*, pages 332–341, March 1995.

[42] V. Ratnakar and M. Livny. An efficient algorithm for optimizing DCT quantization. *IEEE Transaction on Image Processing*, 9:267–270, February 2000.

[43] R. Reininger and J. Gibson. Distributions of the two-dimensional DCT coefficients for images. *IEEE Transaction on Communications*, 31:835–839, June 1983.

[44] J. Rice. *Mathematical Statistics and Data Analysis (Second Edition)*. Duxbury Press,, 1995.

[45] I. E.G. Richardson. *H.264 and MPEG-4 Video Compression*. Wiley, New York, 2003.

[46] M. J. Riely and I. E.G. Richardson. *Digital Video Communications*. Artech House, Boston, 1997.

[47] L. G. Roberts. Picture coding using pseudo-random noise. *IEEE Transaction on Information Theory*, 8:145–154, February 1962.

[48] D. Salomon. *Data Compression: the Complete Reference*. Springer, fourth edition, 2007.

[49] S. Srinivasan, C. Tu, S. L. Regunathan, and G. J. Sullivan. HD Photo: a new image coding technology for digital photography. In *Proceedings of the SPIE Applications of Digital Image Processing XXX*, page Vol. 6696, September 2007.

[50] G. Sullivan. Efficient scalar quantization of exponential and Laplacian random variables. *IEEE Transaction on Information Theory*, 42:1365–1374, September 1996.

[51] G. Sullivan, J. R. Ohm, W.-J. Han, and T. Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22:1649–1668, December 2012.

[52] G. Sullivan and S. Sun. On dead-zone plus uniform threshold scalar quantization. In *Proceedings of the SPIE Visual Communications and Image Processing*, pages 1041–1052, July 2005.

[53] C. Sun and E.-H. Yang. An efficient DCT-based image compression system based on Laplacian transparent composite model. *in preparation for submission to IEEE Transaction on Image Processing*, 2014.

[54] C. Sun and E.-H. Yang. An efficient DCT-based image compression system based on transparent composite model. In *submitted to Proceedings of the IEEE International Conference on Image Processing*, January 2014.

[55] C. Tu, J. Liang, and T. Tran. Adaptive runlength coding. *IEEE Signal Processing Letters*, 10(3):61–64, March 2003.

[56] C. Tu and T. Tran. Context-based entropy coding of block transform coefficients for image compression. *IEEE Transactions on Image Processing*, 11(11):1271–1283, November 2002.

[57] G. K. Wallace. The jpeg still picture compression standard. *ACM Communications*, 34:30–44, April 1991.

[58] G. K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38:18–34, February 1992.

[59] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transaction on Image Processing*, 13:600–612, April 2004.

[60] M. Weinberger, G. Seroussi, and G. Sapiro. The LOCO-I lossless image compression algorithm: principles and standardizations into JPEG-LS. *IEEE Transaction on Image Processing*, 9(8):1309–1324, August 2000.

[61] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:560–576, July 2003.

[62] S.-W. Wu and A. Gersho. Rate-constrained picture-adaptive quantization for JPEG baseline coders. In *Proceedings of the IEEE Acoustics, Speech, and Signal Processing*, pages 389–392, April 1993.

[63] X. Wu and N. Memon. Context-based, adaptive, lossless image coding. *IEEE Transaction on Communications*, 45:437–444, April 1997.

[64] E.-H. Yang. Computational approach to lossy compression and its applications to image/video coding, - a talk at chinese university of Hong Kong, available online at: www.inc.cuhk.edu.hk/seminars/computational-approach-lossy-compression-and-its-applications-imagevideo-coding, November 2010.

[65] E.-H. Yang and Da ke He. Efficient universal lossless compression algorithms based on a greedy sequential grammar transform-part two: With context models. *IEEE Transactions on Information Theory*, 49:2874–2894, November 2003.

[66] E.-H. Yang and J. C. Kieffer. Efficient universal lossless compression algorithms based on a greedy sequential grammar transform-part one: Without context models. *IEEE Transactions on Information Theory*, 46:755–777, May 2000.

[67] E.-H. Yang and C. Sun. Dithered soft decision quantization for baseline JPEG encoding and its joint optimization with Huffman coding and quantization table selection. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, pages 249 –253, November 2011.

[68] E.-H. Yang, C. Sun, and J. Meng. Quantization table design revisited for image/video coding. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1855–1859, September 2013.

[69] E.-H. Yang, C. Sun, and J. Meng. Quantization table design revisited for image/video coding. *IEEE Transaction on Image Processing*, submitted for publication, Jan., 2014.

[70] E.-H. Yang and L. Wang. Full rate distortion optimizaton of MPEG-2 video coding. In *Proceedings of the IEEE International Conference on Image Processing*, pages 605–608, November 2009.

[71] E.-H. Yang and L. Wang. Joint optimization of run-length coding, context-based arithmetic coding and quantization step sizes. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, volume 4, pages 678–681, May 2009.

[72] E.-H. Yang and L. Wang. Joint optimization of run-length coding, Huffman coding, and quantization table with complete baseline JPEG decoder compatibility. *IEEE Transaction on Image Processing*, 18:63–74, January 2009.

[73] E.-H. Yang and X. Yu. Rate distortion optimization for H.264 inter-frame video coding: A general framework and algorithms. *IEEE Transaction on Image Processing*, 16:1774–1784, July 2007.

[74] E.-H. Yang and X. Yu. Soft decision quantization for H.264 with main profile compatibility. *IEEE Transactions on Circuits and Systems for Video Technology*, 19:122–127, January 2009.

[75] E. H. Yang and X. Yu. Transparent composite model for large scale image/video processing. In *Proceedings of the IEEE International Conference on Big Data*, pages 38–44, October 2013.

[76] E.-H. Yang, X. Yu, J. Meng, and C. Sun. Transparent composite model for DCT coefficients: Design and analysis. *IEEE Transaction on Image Processing*, 23:1303–1316, March 2014.

[77] E.-H. Yang and Z. Zhang. The redundancy of source coding with a fidelity criterion-part III: Coding at a fixed distortion level with unknown statistics. *Technical Report, University of Waterloo*, December 1998.

[78] E.-H. Yang and Z. Zhang. Variable-rate trellis source coding. *IEEE Transactions on Information theory*, 45(3):586–608, March 1999.

[79] E.-H. Yang and Z. Zhang. Variable-rate trellis source encoding. *IEEE Transaction on Information Theory*, 45:586–608, March 1999.

[80] E.-H. Yang and Z. Zhang. The redundancy of source coding with a fidelity criterion-part II: Coding at a fixed rate level with unknown statistics. *IEEE Transaction on Information Theory*, 47(1):126–145, January 2001.

[81] E.-H. Yang, Z. Zhang, and T. Berger. Fixed slope universal lossy data compression. *IEEE Transactions on Information theory*, 43(9):1465–1476, September 1997.