Fast Rate Control for JPEG2000 Image Coding

by

Yeung Yick Ming

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy in
Electrical and Electronic Engineering

August 2003, Hong Kong
Fast Rate Control for JPEG2000 Image Coding

by

Yeung Yick Ming

Approved by:

Prof. Oscar C. Au
Thesis Supervisor

Prof. Pengcheng Shi
Thesis Examination Committee Member (Chairman)

Prof. Ross Murch
Thesis Examination Committee Member

Prof. Khaled Ben Letaief
Head of Department

Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
August 2003
Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Yeung Yick Ming
Acknowledgements

I would like to express my sincere appreciation to my thesis supervisor, Dr. Oscar Au. Without his guidance, inspiration and endless enthusiasm, I could not have finished this master thesis.

I would like to thank Dr. Pengcheng Shi and Dr. Ross Murch for serving on my thesis committee, and for their assistance, cooperation, and suggestions.

I would also like to thank my colleague, Andy Chang, for his help in my university life.

Finally, I deeply appreciate my family, my parents Yeung Chi Pui and Tsui Hok Kwun, for their love and support.
Fast Rate Control for JPEG2000 Image Coding

by

Yeung Yick Ming

For the Degree of
Master of Philosophy in Electrical and Electronic Engineering
at The Hong Kong University of Science and Technology
in August 2003

Abstract

JPEG2000 is the new image coding standard which can provide superior rate-distortion performance over the previous JPEG standard. The conventional post-compression rate-distortion (PCRD) optimization scheme in JPEG2000 is not efficient. It requires entropy encoding all available data even though a large portion of them will not be included in the final output. As a result, the entropy coding process can consume up to 60% of the total encoding time. Therefore, reducing its computation can effectively decrease the overall encoding time of JPEG2000.

In this thesis, three fast rate control methods are proposed to efficiently reduce both the computational complexity and memory usage over the conventional PCRD method. The first method, called successive bit-plane rate allocation (SBRA), allocates the bit rate by using the currently available rate-distortion information only. The second method, called priority scanning rate allocation (PSRA), allocates bits according to certain prioritized ordering. The third method uses PSRA to achieve optimal truncation
as PCRD without encoding of all the image details and is called priority scanning with optimal truncation (PSOT).

Simulation results suggest that the three proposed methods provide different trade-off among visual quality, computational complexity, coding delay and working memory size. SBRA is memoryless and casual and requires the least computational complexity, lowest coding delay and achieves good visual quality. PSRA achieves higher PSNR than SBRA at the expense of larger working memory size and longer coding delay. PSOT gives the best PSNR but requires even more computation, delay and memory.
# Table of Contents

Acknowledgements ........................................................................................................ iv
Abstract .......................................................................................................................... v
Table of Contents ........................................................................................................... vii
List of Figures ................................................................................................................ ix
List of Tables ................................................................................................................ xi
Chapter 1 Introduction .................................................................................................. 1
Chapter 2 Overview of JPEG2000 .............................................................................. 5
  2.1 Introduction ............................................................................................................. 5
  2.2 Component Transformations ................................................................................. 6
  2.3 Discrete Wavelet Transform (DWT) ................................................................... 7
     2.3.1 Lifting-based Filtering .................................................................................... 9
  2.4 Quantization .......................................................................................................... 11
  2.5 Entropy Coding (Tier-1 Coding) ....................................................................... 12
  2.6 Rate Control ......................................................................................................... 13
  2.7 Results .................................................................................................................. 14
  2.8 Conclusion ............................................................................................................ 15
Chapter 3 Optimal Rate Control for JPEG2000 .................................................... 16
  3.1 Introduction ............................................................................................................. 16
  3.2 Post Compression Rate Distortion (PCRD) Optimization ............................. 17
     3.2.1 Problems of PQRD ...................................................................................... 20
  3.3 Lagrangian Rate Allocation (LRA) ................................................................. 21
     3.3.1 Problems of LRA ...................................................................................... 23
  3.4 Conclusions ........................................................................................................... 23
Chapter 4 Fast Rate Control for JPEG2000 ........................................................... 25
  4.1 Introduction ............................................................................................................. 25
  4.2 Successive Bit-plane Rate Allocation (SBRA) ............................................... 26
     4.2.1 The Basic Idea ............................................................................................ 27
4.2.2  Determination of $\alpha$ ................................................................................. 29
4.2.3  SBRA Formation ............................................................................................. 32
4.2.3.1 Determination of $\beta$ ................................................................................. 35
4.3  Priority Scanning Rate Allocation (PSRA) ......................................................... 40
4.4  Priority Scanning with Optimal Truncation (PSOT) ........................................... 43
4.5  Simulation Results .............................................................................................. 44
4.6  Conclusions ......................................................................................................... 59
Chapter 5 Conclusions ............................................................................................... 61
5.1  Contributions of This Thesis ............................................................................... 61
5.2  Future Work ........................................................................................................ 62
References .................................................................................................................. 63
List of Publications ..................................................................................................... 66
List of Figures

Fig. 1: Block diagrams of JPEG2000 (a) encoder and (b) decoder. .................................6
Fig. 2: K-level, 1-D wavelet decomposition. The coefficient notation $d_i^j(n)$ refers to the
       $j^{th}$ frequency band (0 for low and 1 for high) of the $i^{th}$ level of decomposition.....8
Fig. 3: Example of two-level 2-D DWT decomposition with subband labels. .........................8
Fig. 4: Example of two-level 2-D DWT decomposition for the test image "barbara" ......8
Fig. 5: Wavelet transform: (a) convolution-based implementation, (b) lifting-based
       implementation ..................................................................................................................10
Fig. 6: Bit-plane and coding pass structure. ........................................................................13
Fig. 7: Visual quality of 'lena' compressed at 0.125bpp using (a) JPEG and (b)
       JPEG2000 .........................................................................................................................14
Fig. 8: Visual quality of 'lena' compressed at 0.25bpp using (a) JPEG and (b) JPEG2000.
       ......................................................................................................................................15
Fig. 9: Rate-distortion performance between optimal and non-optimal rate control .....16
Fig. 10: Lagrangian rate allocation ........................................................................................22
Fig. 11: Block diagram of (a) PCRD and (b) fast rate control framework. .........................25
Fig. 12: Plot of R-D ratio curve of first code-block for test image 'lena'. Bigger dots
       represent bit-plane boundaries. ..........................................................................................32
Fig. 13: Indexing of subbands. Each subband is indexed by a resolution level and an
       orientation (r,q). The lower single index k is computed by (4.24). This example
       shows a three level DWT decomposition ...........................................................................35
Fig. 14: Average rate difference between PCRD and SBRA-1 (SBRA-1–PCRD) for
       sixteen subbands at different target bit-rates .................................................................37
Fig. 15: Test images .................................................................................................................38
Fig. 16: Scan order of (a) subband and (b) code-block within a subband. .........................39
Fig. 17: Average PSNR difference between PCRD and SBRA-w (PCRD–SBRA-w) for
       different values of $\Delta$ at different target bit-rates .......................................................39
Fig. 18: Average rate difference between PCRD and SBRA-w (SBRA-w–PCRD) for
       sixteen subbands at different target bit-rates .................................................................40
Fig. 19: Computational complexity profiles of JPEG2000 using PCRD in test images (a) 'barbara', (b) 'aerial2', (c) 'bike' and (d) 'tools'.
Fig. 20: Average computational complexity profile of JPEG2000 using PCRD in the fifteen test images in Fig. 15.
Fig. 21: Average number of CPU clock cycles per pixel required for tier-1 coding for the test images in Fig. 15.
Fig. 22: Average working memory size for the test images in Fig. 15.
Fig. 23: Average PSNR difference between proposed methods and PCRD (proposed−PCRD) for the test images in Fig. 15.
Fig. 24: Visual quality of 'lena' compressed at 0.125bpp using (a) PCRD (30.98dB), (b) SBRA-1 (30.84dB), (c) SBRA-w (30.85dB) and (d) PSRA (30.90dB).
Fig. 25: Visual quality of 'lena' compressed at 0.25bpp using (a) PCRD (34.11dB), (b) SBRA-1 (33.78dB), (c) SBRA-w (33.87dB) and (d) PSRA (34.03dB).
Fig. 26: Visual quality of 'lena' compressed at 1bpp using (a) PCRD (40.38dB), (b) SBRA-1 (39.38dB), (c) SBRA-w (39.82dB) and (d) PSRA (40.34dB).
Fig. 27: Average rate control accuracy for the test images in Fig. 15.
List of Tables

Table 1: Daubechies 9/7 analysis and synthesis filter coefficients.................................9
Table 2: 5/3 analysis and synthesis filter coefficients.................................................9
Table 3: Complexity comparison of convolution and lifting-based implementation. ....11
Table 4: Computational complexity results, measured in number of CPU clock cycles per pixel required for tier-1 encoding. (Profile at PIII-800 384M RAM, Intel Vtune performance analyzer 6.1 and Windows XP).................................................48
Table 5: Working memory size results, measured in bits per pixel.............................50
Table 6: PSNR results, measured in dB.......................................................................52
Chapter 1

Introduction

With the increasing use of multimedia and network technologies, the efficient use of channel bandwidth is one of the main issues in transmitting image/video data. Rate control or rate allocation is necessary to control the bit-rate of image/video coding such that it meets the channel bandwidth, end-to-end delay or storage requirement. The ultimate goal of rate control is to allocate the target bit-rate into an image/video such that the overall distortion can be minimized.

This thesis explores the rate control techniques in JPEG2000. JPEG2000 [1-5] is a new international standard for still image coding. It can provide both objective and subjective image quality superior to existing standards such as JPEG [6, 7]. The JPEG2000 coder comprises discrete wavelet transform (DWT) and bit-plane MQ coder [8]. In the conventional discrete cosine transform (DCT) based JPEG standard, the bit-rate is controlled by a single value of quantization factor (or quality factor). However JPEG can only control bit-rate by choosing different quantization factors which requires re-quantization and re-entropy coding of the coefficient data. By using the bit-plane coding, JPEG2000 can control the bit-rate to meet the bit-rate requirement precisely and easily. In Chapter 2, we will provide an overview of JPEG2000.

The basic encoding algorithm of JPEG2000 is based on EBCOT (Embedded Block Coding with Optimized Truncation) [8]. The algorithm partitions the wavelet
coefficient into non-overlapping rectangle blocks called code-blocks in JPEG2000. The code-block data are then entropy encoded by bit-plane coding. A rate-distortion optimization (optimal bit allocation) process is applied after all the quantized wavelet coefficients have been entropy encoded (compressed) and is referred to as post-compression rate-distortion (PCRD) optimization [8, 9]. By utilizing the actual rate-distortion functions of all compressed data, the PCRD technique attains the minimum image distortion for a given bit-rate. However, since it requires encoding all the data and storing all the encoded bit-stream even though a large portion of the data needs not to be sent out, most of the computation and memory usage could be redundant in this process. Also the PCRD is an off-line process such that the whole image needs to be completely encoded before sending out any data and hence long delay is possible.

Another technique for the JPEG2000 optimal rate allocation problem is by coefficients modeling. Kasner et al. [10] assumed that the wavelet coefficients could be modeled by memoryless generalized-Gaussian density (GGD). By estimating the GGD parameter, the rate-distortion function can be approximated as required for the optimal rate allocation. This approach is included in Part-2 of JPEG2000 [2] and is called Lagrangian rate allocation (LRA). In this approach, both the rate and distortion are estimated before the wavelet coefficients are actually encoded. A quantization step size of each subband is selected based on the estimation and the quantized wavelet coefficients are encoded without any truncation. However the rate control accuracy depends heavily on the correctness of the GGD model assumption and the resulting bit-rate can be different from the target value. LRA is thus an iterative technique which takes typically many iterations to achieve the target bit-rate. In each iteration, the quantization step sizes are required to be re-estimated and the wavelet coefficients are
thus re-quantized and entropy re-encoded again. The multiple quantization and entropy encoding processes heavily increase the complexity of this approach. In practice, the complexity of LRA is comparable to the PCRD approach. In Chapter 3, we will investigate both the PCRD and LRA approaches in JPEG2000.

Other than the empirical PCRD approach and the analytical LRA approach, Masuzaki et al. [11] first proposed a non-optimal training-image based fast rate control method for JPEG2000. By applying PCRD to a set of training images, the proposed fast method learns the relationship between the number of coding passes (coding points) and the corresponding number of bytes within a subband. The relationship is then approximated by a linear curve. Given a target bit-rate, the fast method predicts the number of coding passes to be included in the final output using the linear model. However the paper shows that the method can suffer from a significant Peak Signal-to-Noise Ratio (PSNR) loss (more than 1dB at 0.25bpp). The loss could be much more as the linear model may not be good for all images. The PSNR is computed as

\[
PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) dB
\]  

where MSE is mean square error between the original and reconstructed images.

Model based rate allocation is an attractive approach for fast rate control as it can provide the optimal quality when the coefficients follow the model assumption. However the major drawback is the degree of model accuracy. It is unlikely that we can find a model that is good for all images. Thus we focus on non-model based or predictive-based fast rate control. In Chapter 4, we propose three new fast rate control methods that can efficiently reduce or remove the computation and memory usage redundancy over the conventional PCRD method. The first method, called successive
bit-plane rate allocation (SBRA), assigns the maximum allowable bit-rate for each code-block’s bit-plane by using the currently available rate-distortion information only. The second method is called priority scanning rate allocation (PSRA). It first predicts how the truncation points can be arranged in descending order of rate-distortion slope and then encodes the truncation points based on the order (priority) information. The third method uses PSRA to obtain a smaller amount of data for optimal truncation and is called priority scanning with optimal truncation (PSOT). Among the three methods, SBRA achieves the maximum reduction in computational complexity and memory usage, and the lowest coding/transmission delay. The computational complexity reduction can be up to about 90% of the entropy coding process. However it gives the worst PSNR performance. PSRA achieves higher PSNR than SBRA with the penalty of lower memory usage reduction and longer delay. PSOT achieves the best (optimal) quality while being the least efficient method in term of computational complexity, memory usage and the coding/transmission delay. The three proposed methods provide different trade-off of computational complexity, memory reduction, coding/transmission delay and PSNR performance.
Chapter 2

Overview of JPEG2000

2.1 Introduction

JPEG2000 is the new international standard for still image coding. In the following, we will focus on the Part I [1] (or baseline) of the standard that defines the core coding system. The block diagram of the JPEG2000 encoder is illustrated in Fig. 1a. All the image samples are first DC level shifted by subtracting the same quantity (i.e. subtracted by 128 for 8-bit image). After that, a forward component (color) transform is applied. Then the forward discrete wavelet transform (DWT) decomposes each of the image components into subbands with one or more levels (typically five levels in JPEG2000) of decomposition. The subbands consist of coefficients that represent certain horizontal and vertical spatial frequency characteristics of the image. Each subband is then quantized by a scalar quantizer and divided into non-overlapped rectangular blocks called code-blocks, with typical size of 64×64. The quantized wavelet coefficients in a code-block are entropy encoded to form a code-block bit-stream. Each of the code-block bit-stream can be truncated by rate control to meet the target bit-rate. The collection of the truncated code-block bit-streams will be packaged into data units called packets and transmitted to the channel. This final output is called code-stream in JPEG2000. The
decoder (Fig. 1b) is the reverse of the encoder.

![Block diagrams of JPEG2000](image)

Fig. 1: Block diagrams of JPEG2000 (a) encoder and (b) decoder.

### 2.2 Component Transformations

For the input image, it can be a grey-scale (single component) image or color (three components) image. As grey-scale image has only one component, there will be no component transformation for grey-scale image. In color image, the input format is, in general, the Red-Green-Blue (RGB) format.

JPEG2000 supports two types of component transformations. One is *irreversible component transformation* (ICT) that can be used for lossy compression only. Another one is *reversible component transformation* (RCT) that can be used for lossy or lossless compression. The forward and inverse ICT are achieved by (2.1a) and (2.1b) respectively.

\[
\begin{bmatrix}
Y \\
C_y \\
C_r
\end{bmatrix} =
\begin{bmatrix}
0.299 & 0.587 & 0.144 \\
-0.16875 & -0.33126 & 0.5 \\
0.5 & -0.41869 & -0.08131
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]  
(2.1a)
\[
\begin{pmatrix}
R \\ G \\ B
\end{pmatrix} =
\begin{pmatrix}
1.0 & 0 & 1.402 \\
1.0 & -0.34413 & -0.71414 \\
1.0 & 1.772 & 0
\end{pmatrix}
\begin{pmatrix}
Y \\ C_b \\ C_r
\end{pmatrix}
\] (2.1b)

The forward and inverse RCT are performed by (2.2a) and (2.2b) respectively.

\[
\begin{pmatrix}
Y_r \\ U_r \\ V_r
\end{pmatrix} =
\begin{pmatrix}
R + 2G + B \\ 4 \\ R - G \\ B - G
\end{pmatrix}
\] (2.2a)

\[
\begin{pmatrix}
R \\ G \\ B
\end{pmatrix} =
\begin{pmatrix}
Y_r - \frac{U_r + V_r}{4} \\ U_r + G \\ V_r + G
\end{pmatrix}
\] (2.2b)

2.3 Discrete Wavelet Transform (DWT)

Instead of using the conventional discrete cosine transform (DCT), JPEG2000 uses the wavelet transform [12-15] as the transformation kernel. The generic form for a one-dimensional (1-D) DWT is shown in Fig. 2. A signal is first filtered by a low-pass and high-pass filter, \( h \) and \( g \), respectively, and then down sampled by a factor of two.

We call the forward and inverse transform filter as "analysis filter" and "synthesis filter" respectively. Repeating the filtering and down-sampling process on the low-pass branch outputs make multiple levels of the wavelet decomposition. The 1-D DWT can be extended to a two-dimensional (2-D) DWT by applying a 1-D transform to all the rows of the input, and then repeating on all of the resulting columns. The forward 2-D DWT decomposes an image into subbands with one or more levels of decomposition. Fig. 3 shows an example of two-level 2-D DWT decomposition with subband labels. An
example of two-level 2-D DWT decomposition for the test image 'barbara' is shown in Fig. 4.

![Diagram of two-level 2-D DWT decomposition](image)

Fig. 2: K-level, 1-D wavelet decomposition. The coefficient notation \(d_{ij}(n)\) refers to the \(j^{th}\) frequency band (0 for low and 1 for high) of the \(i^{th}\) level of decomposition.

![Diagram of subband labels](image)

Fig. 3: Example of two-level 2-D DWT decomposition with subband labels.

![Example of two-level 2-D DWT decomposition](image)

Fig. 4: Example of two-level 2-D DWT decomposition for the test image 'barbara'.

The DWT can be irreversible or reversible. The irreversible transform is used for lossy compression while the reversible transform is for lossless compression. The
default irreversible transform is performed by the Daubechies 9/7 filter [16]. Both the analysis and the synthesis filter coefficients are given in Table 1. The default reversible transform is performed by the 5/3 filter [17-19]. The coefficients of the 5/3 filter are given in Table 2.

<table>
<thead>
<tr>
<th>i</th>
<th>Lowpass Filter $h_a(i)$</th>
<th>Highpass Filter $g_a(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.6029490182363579</td>
<td>1.115087052456994</td>
</tr>
<tr>
<td>±1</td>
<td>0.2668641184428723</td>
<td>-0.5912717631142470</td>
</tr>
<tr>
<td>±2</td>
<td>-0.0782326652898785</td>
<td>-0.05754352622849957</td>
</tr>
<tr>
<td>±3</td>
<td>-0.01686411844287495</td>
<td>0.09127176311424948</td>
</tr>
<tr>
<td>±4</td>
<td>0.02674875741080976</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>Lowpass Filter $h_a(i)$</th>
<th>Highpass Filter $g_a(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.115087052456994</td>
<td>0.6029490182363579</td>
</tr>
<tr>
<td>±1</td>
<td>0.5912717631142470</td>
<td>-0.2668641184428723</td>
</tr>
<tr>
<td>±2</td>
<td>-0.05754352622849957</td>
<td>-0.0782326652898785</td>
</tr>
<tr>
<td>±3</td>
<td>-0.09127176311424948</td>
<td>0.01686411844287495</td>
</tr>
<tr>
<td>±4</td>
<td>0.02674875741080976</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Daubechies 9/7 analysis and synthesis filter coefficients

<table>
<thead>
<tr>
<th>i</th>
<th>Analysis Filter Coefficients</th>
<th>Synthesis Filter Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lowpass Filter $h_a(i)$</td>
<td>Highpass Filter $g_a(i)$</td>
</tr>
<tr>
<td>0</td>
<td>6/8</td>
<td>1</td>
</tr>
<tr>
<td>±1</td>
<td>2/8</td>
<td>-1/2</td>
</tr>
<tr>
<td>±2</td>
<td>-1/8</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: 5/3 analysis and synthesis filter coefficients

2.3.1 Lifting-based Filtering

Each level of the wavelet decomposition shown in Fig. 2 is in the form of convolution-based filtering in which the filter output is the dot product between the filter mask and the 1-D signal. However the standard supports another mode of filtering which is called lifting-based filtering [20, 21, 22]. The basic principle of the lifting scheme is to
factorize the wavelet filter into a sequence of alternating upper and lower triangular matrices and a diagonal matrix with constant elements. Fig. 5 shows both the convolution-based and lifting-based implementation of wavelet transform.

![Wavelet Transform Diagrams](image)

**Fig. 5:** Wavelet transform: (a) convolution-based implementation, (b) lifting-based implementation

The lifting scheme consists of three stages: *split*, *predict* and *update*. In the first stage, the 1-D signal is split into an even indexed and an odd indexed signal. In the second stage, the even signal is filtered by a prediction filter and the highpass output $y_1[n]$ is the scaled difference between the filtered even signal and the odd signal. In the third stage, the difference (or residue) signal in the second stage is filtered by an updating filter and added back to the even signal. The lowpass output $y_0[n]$ is then the scaled version of the updated even signal.

There are two significant advantages of the lifting scheme. First, the computational complexity of lifting-based implementation is lower than the convolution-based implementation. Table 3 shows the complexity comparison of convolution and
lifting-based implementation for both 5/3 and 9/7 filters. The second advantage is that
lifting provides in-place computation of wavelet coefficients by overwriting the memory
locations which contain the input sample values. This can greatly reduce the memory
usage. Because of these advantages, the standard uses lifting as the default filtering
mode.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Multiplications</th>
<th>Additions</th>
<th>Multiplications</th>
<th>Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/3</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>9/7</td>
<td>9</td>
<td>14</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 3: Complexity comparison of convolution and lifting-based implementation.

2.4 Quantization

After the DWT, the wavelet coefficients are quantized using scalar quantization.

Each of the coefficients $a_b(x, y)$ of the subband $b$ is quantized to the value $q_b(x, y)$ by

$$q_b(x, y) = \text{sign}(a_b(x, y)) \cdot \frac{|a_b(x, y)|}{\Delta_b}$$  \hspace{1cm} (2.3)

where $\Delta_b$ is the quantization step size of the subband $b$.

In lossless compression, the standard requires the value of $\Delta_b$ to be one for all
subbands. However, in lossy compression, no particular selection of the quantization
step size is required in the standard. One way in selecting the quantization step size is to
scale a default (or pre-defined) step size $\Delta_d$ by an energy weight parameter $\gamma_b$ [23] as
follows

$$\Delta_b = \frac{\Delta_d}{\sqrt{\gamma_b}}$$  \hspace{1cm} (2.4)
This selection of quantization step size is recommended in the standard and is implemented in the standard reference software [26, 27, 28] with the default step size \( \Delta_d \) equal to two for all subbands.

2.5 Entropy Coding (Tier-1 Coding)

The quantized wavelet coefficients in the code-blocks are encoded using coefficient bit modeling and arithmetic coding. This process is called tier-1 coding in JPEG2000. Tier-1 coding is essentially a bit-plane coding technique which is commonly used in wavelet based image coders [24, 25]. In tier-1 coding, code-blocks are encoded independently. For each code-block, coefficients are encoded starting from the most significant bit-plane (MSB) with a non-zero element towards the least significant bit-plane (LSB). Each coefficient bit in a bit-plane is selected to be included in only one of the three coding passes called significance pass, refinement pass and cleanup pass by using coefficient bit modeling. The coding pass data are then arithmetic encoded by a context-based adaptive binary arithmetic coder, which is called MQ-coder in JPEG2000. The MQ-coder is also used in JBIG2 standard [29]. The bit-plane and coding pass structure is shown in Fig. 6. Except for the MSB, where it includes only one coding pass (cleanup pass), each of the other bit-planes consists of three coding passes. The encoded coding pass data will form a bit-stream which is organized as shown in Fig. 6.
2.6 Rate Control

Rate control in JPEG2000 is achieved by both the quantization and selection of the coding pass data to be included in the code-stream. The quantization process as mentioned before is applied once to roughly control the bit-rate. The resulting bit-rate is usually far from the target bit-rate. The accurate rate control is achieved by the selection of the coding pass data of each code-block to be included in the final code-stream. In other words, the code-block bit-stream will be truncated at a particular point, and discard all the coding passes behind that point. JPEG2000 has no requirement on the selection of a particular rate control method. However an optimal rate control process called post-compression rate-distortion (PCRD) optimization is recommended in the standard and we simply call it as PCRD. This method will be described in Chapter 3.

Fig. 6: Bit-plane and coding pass structure.
Another optimal rate control method is called Lagrangian rate allocation (LRA) in which the bit-rate is controlled by choosing the quantization step sizes only. This method is mentioned in Part 2 of the standard [2] and we will briefly describe it in Chapter 3.

2.7 Results

To check the superiority of JPEG2000, the image ‘lena’ is compressed by JPEG and JPEG2000 for different bit-rates. The visual quality comparisons of ‘lena’ compressed at 0.125bpp (bit per pixel) and 0.25bpp are shown in Fig. 7 and Fig. 8 respectively. For a large category of images, [30] had shown that JPEG2000 file sizes were on average 18% smaller than JPEG at 0.75bpp, 36% smaller at 0.5bpp and 53% smaller at 0.25bpp.

Fig. 7: Visual quality of 'lena' compressed at 0.125bpp using (a) JPEG and (b) JPEG2000.
Fig. 8: Visual quality of 'lena' compressed at 0.25bpp using (a) JPEG and (b) JPEG2000.

2.8 Conclusion

In this chapter, we have given the overview of JPEG2000. The coding architecture of JPEG2000 is very different from the conventional JPEG standard. Instead of using the traditional DCT, JPEG2000 uses DWT as the transform method. As a result, the coding efficient of JPEG2000 is much higher than JPEG. However, JPEG2000 is more complicated than JPEG and thus a fast algorithm in implementing the building blocks of JPEG2000 is often desired.
Chapter 3

Optimal Rate Control for JPEG2000

3.1 Introduction

The goal of optimal rate control is to minimize the distortion of the reconstructed image for a given rate constraint. For a particular coding method such as JPEG2000, the optimal rate control is to achieve the rate-distortion performance which lies on the lower bound or convex hull in all possible set of rate-distortion (R-D) curves. Fig. 9 illustrates the rate-distortion performance achieved by optimal and non-optimal rate control.

![Rate-distortion performance between optimal and non-optimal rate control](image)

Fig. 9: Rate-distortion performance between optimal and non-optimal rate control.
There are two types of optimal rate control processes. The first type uses the actual R-D information of the sample data. This type of optimal rate control can be referred to as post compression rate distortion (PCRD) optimization and it is recommended as the rate control process in baseline JPEG2000. Instead of using the actual R-D information, the second type of optimal rate control models the R-D relationship of the sample data by a prior assumption of the sample data. This type of optimal rate control is called model-based optimization. One model-based optimal rate control process is mentioned in Part 2 of the standard [2]. As it is not a recommended baseline rate control method, we will only briefly describe it in this chapter.

3.2 Post Compression Rate Distortion (PCRD) Optimization

The post compression rate distortion optimization is a recommended rate control method for the baseline JPEG2000. We simply call this method as PCRD. This process had been described in [8] clearly and we will summarize it as follows.

Let \( \{ B_i \}_{i=1,2,...} \) denote the set of code-blocks in the whole image. For each code-block, an embedded bit-stream is formed by the tier-1 coding of all the bit-planes from MSB to LSB. In the bit-stream, there is a set of feasible truncation points each of which is defined at the end of a coding pass. In this thesis, we use \( n_i \) to identify the feasible truncation points of the \( i^{th} \) code-block \( B_i \), with \( n_i = k \) corresponding to the \( k^{th} \) truncation point from the MSB. For code block \( B_i \), the bit-stream can be truncated at any feasible truncation point \( n_i \), resulting in corresponding discrete length or bit-rate.
The corresponding distortion incurred by reconstructing the truncated bit-stream is denoted by $D_i^n$. The distortion $D_i^n$ is computed by following equation:

$$D_i^n = w_b^2 \sum_{k \in B_i} (\hat{s}_k^n[k] - s_k[k])^2$$ (3.1)

Here $s_k[k]$ is the 2-D sequence of subband samples in code-block $B_i$, $\hat{s}_k^n[k]$ is the reconstructed subband samples with truncation point $n_i$, and $w_b^2$ is the energy of the wavelet basis functions for subband $b$ which code-block $B_i$ belongs to.

The rate control optimization process selects the truncation points of all code-blocks to minimize the overall reconstructed image distortion $D$ where

$$D = \sum_i D_i^n$$ (3.2)

subject to the rate constraint

$$R = \sum_i R_i^n \leq R_{\text{budget}}$$ (3.3)

where $R_{\text{budget}}$ denotes the target bit-rate. This constrained optimization problem is solved by the Lagrange multiplier technique [8, 9]. The optimization process is then equivalent to minimize the cost function

$$J = D + \lambda R = \sum_i (D_i^n(\lambda) + \lambda R_i^n(\lambda))$$ (3.4)

Therefore if we can find a value of $\lambda$ such that the resulting set of truncation points $\{n_i(\lambda)\}_{i=1,2,...}$ minimizes (3.4) and yields $R = R_{\text{budget}}$, both the value of $\lambda$ and the set of truncation points will be optimal in the sense that we cannot reduce the distortion without increasing the bit-rate beyond $R_{\text{budget}}$.

PCRD [8] is a simple algorithm to find the optimal truncation points. At any
feasible truncation point $n_i$, PCRD computes the R-D "slope" which is defined as

$$S^i_{n} = \frac{\Delta D^i_{n}}{\Delta R^i_{n}} = \frac{D^i_{n-1} - D^i_{n}}{R^i_{n} - R^i_{n-1}}$$

(3.5)

In the rest of the thesis, the term R-D slope is always used to refer to (3.5). Let $N_i$ be the set of all feasible truncation points of code-block $B_i$. The truncation point $z_i(\lambda)$ for any value of $\lambda$ is found in PCRD by

$$z_i(\lambda) = \max \{j \in N_i \mid S^j_i \geq \lambda\}$$

(3.6)

Assuming that the R-D slope is strictly decreasing [8] such that $S^{n+1}_i < S^n_i$ for any feasible truncation point $n_i$, the optimal value of $\lambda$ denoted as $\lambda_{\text{optimal}}$ is equal to the minimum value of $\lambda$ (or R-D slope) which satisfies the rate constraint in (3.3). Theoretically, there are infinitely many possible values of $\lambda$. Thus an iterative approach with fast convergence is used in PCRD to search for the $\lambda_{\text{optimal}}$. Once we know the $\lambda_{\text{optimal}}$, the optimal truncation points can be found by (3.6) with $\lambda = \lambda_{\text{optimal}}$. The algorithm of PCRD is basically:

1. Encode all the code-blocks $B_i$ fully (including all the feasible truncation point $n_i$) for all $i$.
2. Compute $\lambda_{\text{optimal}}$ using the actual R-D slope information of all the code-blocks.
3. Initialize $i = 1$.
4. Initialize $n_i = 0$.
5. Set $n_i = n_i + 1$. Compute $S^n_i$. 

19
6. If $S_i^h \geq \lambda_{\text{optimal}}$, go to step 5. Else choose $z_i = n_i - 1$ as the selected truncation point for code-block $B_i$.

7. If $B_i$ is not the last code-block, set $i = i + 1$ and go to step 4.

However, the R-D slopes at the feasible truncation points of real images may not be strictly decreasing, especially those in the initial few bit-planes. Thus, in the PCRD implementation, the feasible truncation points at which the R-D slopes are not strictly decreasing are considered 'unfeasible' and PCRD would not truncate at those points.

### 3.2.1 Problems of PCRD

In PCRD, the R-D slope information at all the feasible truncation points are pre-computed and stored in memory. This requires tier-1 encoding of all the quantized coefficients and the storage of the whole encoded bit-stream in memory buffer even though a large portion of them will not be included in the code-stream after the optimal truncation. Therefore a significant portion of computational power and working memory size is wasted on computing and storing the unused data. We call this portion of wasted computational power and working memory size to be redundant computation and redundant memory usage respectively. Also PCRD is a non-causal or off-line process because the entire image needs to be completely encoded before the code-stream can be determined and sent out to channel. Hence long transmission delay is possible. Since PCRD requires tier-1 encoding of all the quantized coefficients, according to the profiles reported by [31], its computational complexity could be up to 60% of the total CPU execution time.
3.3 Lagrangian Rate Allocation (LRA)

Lagrangian rate allocation (LRA) is a model-based optimal rate control method. This method is originally proposed in [10] and included in Part 2 of the standard [2] as an alternative rate control method. Instead of selecting the truncation points, this method controls the bit-rate only by choosing the quantization step size for each subband. After choosing the quantization step size, all the quantized coefficients are included in the final code-stream without truncation. Here we just give the basic idea behind LRA. Reader can refer to [10] for the mathematical detail. This method can be summarized as follows.

LRA first models wavelet coefficients by memoryless generalized-Gaussian density (GGD) which is given by

\[
p(x) = \frac{\alpha}{2\sigma \Gamma(1/\alpha)} \sqrt{\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}} \exp \left\{-\left(\frac{\Gamma(3/\alpha)}{\sqrt{\Gamma(1/\alpha)}} \frac{|x|}{\sigma}\right)^\alpha\right\}
\]

(3.7)

where \( x \) is the wavelet coefficients, \( \sigma \) is the standard deviation of \( x \) and \( \alpha \) is the GGD parameter. Using the Lagrange multiplier technique, the rate of each subband \( b \) is computed by

\[
R_b = g_b \left( \frac{\lambda}{\gamma_b \sigma_b^2} \right)
\]

(3.8)

where \( \lambda \) is a Lagrange multiplier, \( \gamma_b \) is the energy of the wavelet basis functions for subband \( b \) and \( \sigma_b \) is the standard deviation of subband samples. The function \( g_b \) is then modeled by the GGD model. Once the rate information has been computed, the quantization step sizes required achieving those rates is modeled by a function \( f_b \) such that

\[
\Delta_b = f_b(R_b)
\]

(3.9)
In Part 2 of the standard, the LRA process is defined in Fig. 10.

\[ R_D : \text{Target Bit-rate} \]
\[ R_A : \text{Achieved Bit-rate} \]
\[ D1: R' \text{ within tolerance of } \bar{R}_D \? \]
\[ D2: R_A \text{ within tolerance of } R_D \? \]

Fig. 10: Lagrangian rate allocation.
3.3.1 Problems of LRA

In this approach, both the rate and distortion are estimated before the wavelet coefficients are actually encoded. A quantization step size of each subband is selected based on the estimation and the quantized wavelet coefficients are encoded without any truncation. However the rate control accuracy depends heavily on the correctness of the GGD model assumption and the resulting bit-rate can be different from the target value. In practice, the resulting bit-rate may be 10-20% different from the target bit-rate. That is the reason why the LRA use an iterative approach to converge to the target bit-rate as shown in Fig. 10. In each iteration, the quantization step sizes are required to be re-estimated and the wavelet coefficients are thus re-quantized and entropy re-encoded again. The multiple quantization and entropy encoding processes heavily increase the complexity of this approach. In practice, the complexity of LRA is comparable to the PCRD approach.

As LRA controls the bit-rate by only one quantization step size for each subband, it is not convenient to perform SNR (or quality) scalability. SNR scalability means that image can be progressively recovered by quality. In PCRD scheme, the rate is controlled by selecting the truncation points of code-block bit-streams among a large set of feasible truncation points. It is much more efficient in performing SNR scalability.

3.4 Conclusions

In this chapter, we have introduced two types of optimal rate control methods for JPEG2000. PCRD uses the actual rate-distortion information after compressing (or entropy encoding) the sample coefficients. This method can provide the real optimal
rate-distortion performance. Another method, LRA, models the rate-distortion function of subband coefficients prior to the compression. As LRA does not use the actual rate-distortion information, it may provide the real optimal rate-distortion performance only when the subband coefficients follow the GGD assumption. Thus we can say that the LRA is a sub-optimal rate control method.

PCRD can provide the optimal rate-distortion performance. However it is not efficient in term of computational complexity. By using the PCRD method, the entropy encoding can become the most computational intensive part in JPEG2000. It would be highly desirable to develop a more efficient rate control method to reduce the encoding complexity and encoding delay.
Chapter 4

Fast Rate Control for JPEG2000

4.1 Introduction

We have mentioned, in previous chapter, two optimal rate control methods for JPEG2000. The problem of those optimal methods is that the computational complexity is high. In this chapter, we will propose three fast rate control methods for JPEG2000.

As selecting the truncation points is a recommended rate control method for the baseline JPEG2000, we will restrict our discussion and comparison to PCRD only. In PCRD, the rate control is performed after finishing the entire tier-1 encoding process. To have an early termination of the tier-1 encoding process, the proposed three fast rate control methods use a feedback control as shown in Fig. 11.

![Block diagrams](attachment:diagram.png)

Fig. 11: Block diagram of (a) PCRD and (b) fast rate control framework.
Using the feedback control, the three fast rate control methods can efficiently reduce both the computational complexity and memory usage over PCRD. The first method, called successive bit-plane rate allocation (SBRA), allocates the bit rate by using the currently available rate-distortion information only. The second method performs rate control by predicting the order of magnitude of each truncation point’s rate-distortion slope. We called the second method as priority scanning rate allocation (PSRA). The third method uses PSRA to obtain a smaller amount of data set for optimal truncation and is called priority scanning with optimal truncation (PSOT). The proposed three methods provide different coding/transmission delay, degree of complexity reduction and PSNR performance.

4.2 Successive Bit-plane Rate Allocation (SBRA)

PCRD, as mentioned in Chapter 3, incurs significant encoding delay because it selects the truncation points only after the completion of tier-1 encoding of all the code-blocks. For a real-time, low-delay rate control process, it is desirable that the compressed data of the early code-blocks can be sent out before the later (or future) code-blocks are being encoded. The ideal way to do this is to select the truncation point of a code-block in parallel with the tier-1 coding. This can also eliminate the redundant computation and memory usage. However without knowing the actual R-D slopes of the future code-block data, it is impossible to find the optimal truncation point. So instead of finding the optimal truncation point, we propose to find a good truncation point that is close to the optimal point and can be computed in parallel with the tier-1 coding.
4.2.1 The Basic Idea

Recall that the optimal truncation point \( z_i \) is selected such that

\[
S_i^{n_i} = \frac{\Delta D_i^{n_i}}{\Delta R_i^{n_i}} = \frac{D_i^{n_i-1} - D_i^{n_i}}{R_i^{n_i} - R_i^{n_i-1}} \geq \lambda_{\text{optimal}}
\]

(4.1)

for \( n_i \leq z_i \). In particular, \( S_i^{z_i} \geq \lambda_{\text{optimal}} \) with \( S_i^{z_i} = \lambda_{\text{optimal}} \). As \( \lambda_{\text{optimal}} \) cannot be computed in a casual manner, PCRD is not casual. We will now propose a casual expression similar to \( S_i^{n_i} \) which can be compared to a casual threshold. We define \( A_i^{z_i:n_i} \), for \( n_i < z_i \), as

\[
A_i^{z_i:n_i} = \frac{D_i^{n_i}}{R_i^{z_i} - R_i^{n_i}}
\]

(4.2)

As \( z_i \) is the final selected truncation point of the \( i^{th} \) code-block \( B_i \) such that \( R_i^{n_i} < R_i^{z_i} \) for any \( n_i < z_i \), \( A_i^{z_i:n_i} \) is a positive quantity for \( n_i < z_i \). \( A_i^{z_i:n_i} \) and \( S_i^{n_i} \) are related by

\[
S_i^{n_i} = \frac{D_i^{n_i-1}}{R_i^{n_i} - R_i^{n_i-1}} + \frac{D_i^{n_i}}{R_i^{n_i} - R_i^{n_i-1}} = A_i^{z_i:n_i} - \frac{D_i^{n_i}}{R_i^{n_i} - R_i^{n_i-1}}
\]

(4.3)

\[
S_i^{n_i} \geq \lambda_{\text{optimal}} \iff A_i^{z_i:n_i} \geq \lambda_{\text{optimal}} + \frac{D_i^{n_i}}{R_i^{n_i} - R_i^{n_i-1}}
\]

(4.4)

and at the truncation point \( z_i \)

\[
S_i^{z_i} \geq \lambda_{\text{optimal}} \Rightarrow A_i^{z_i:n_i} \geq \lambda_{\text{optimal}} + \frac{D_i^{z_i}}{R_i^{z_i} - R_i^{z_i-1}}
\]

(4.5)

Here we have defined a new quantity \( \alpha_i^{z_i} \). We call the term \( A_i^{z_i:n_i} \) as rate-distortion ratio (R-D ratio). Note that \( A_i^{z_i:n_i} > A_i^{z_i:n_i} \) for any \( n_i < z_i \) because \( R_i^{n_i} < R_i^{z_i} \). The R-D ratio will be used in the proposed SBRA algorithm. We assume that \( A_i^{z_i:n_i} \) is strictly
decreasing with respect to $n_i$ for $n_i < z_i$. This implies that, among all the $n_i < z_i$, the minimum value of $A^{n_i \cdot n_i - 1}_i$ is $A^{z_i \cdot z_i - 1}_i$. Thus we can show that $A^{n_i \cdot n_i - 1}_i > A^{z_i \cdot z_i - 1}_i > A^{z_i \cdot z_i - 1}_i \geq A^{z_i \cdot z_i - 1}_i \geq \alpha_i^{z_i}$ as

$$A^{n_i \cdot n_i - 1}_i > A^{z_i \cdot z_i - 1}_i > A^{z_i \cdot z_i - 1}_i \geq A^{z_i \cdot z_i - 1}_i \geq \alpha_i^{z_i}$$

(4.6)

for $n_i < z_i$. If it happens that $A^{n_i \cdot n_i - 1}_i \leq \alpha_i^{z_i}$, then $n_i$ has been already on or beyond the truncation point ($n_i \geq z_i$). This is the basic idea of SBRA. Assuming the ideal situation that $\alpha_i^{z_i}$ is available, the proposed SBRA-ideal (ideal form of SBRA) algorithm is:

1. Initialize $i = 1$.
2. Initialize $n_i = 0$.
3. Set $n_i = n_i + 1$. Encode the $n_i^{th}$ coding pass of code-block $B_i$ (i.e. encode $B_i$ up to the feasible truncation point $n_i$).
4. Assume $n_i = z_i$ and compute both $A^{n_i \cdot n_i - 1}_i$ and $\alpha_i^{z_i} \equiv \lambda_{optimal} + \frac{D_i^{n_i}}{R_i^{n_i} - R_i^{n_i - 1}}$.
5. If $A^{n_i \cdot n_i - 1}_i \geq \alpha_i^{z_i}$, go to step 3. Else choose $z_i = n_i - 1$ as the selected truncation point.
6. If $B_i$ is not the last code-block, set $i = i + 1$ and go to step 2.

The structure of SBRA-ideal is very different from PCRD because the code-block encoding is no longer performed at the very beginning. Instead, it is performed incrementally in step 3, with one coding pass at a time. In step 4, the assumption of $n_i = z_i$ implies that $A^{n_i \cdot n_i - 1}_i \geq \alpha_i^{z_i}$ should be true because of (4.6). If the test $A^{n_i \cdot n_i - 1}_i \geq \alpha_i^{z_i}$ in step 5 is indeed true, $n_i$ may or may not be the desired truncation point because $A^{n_i \cdot n_i - 1}_i \geq \alpha_i^{z_i}$ is a necessary but not sufficient condition for $n_i = z_i$. As it is not
conclusive, SBRA-ideal would examine another feasible truncation point. On the other hand, if\( A_i^{n_i-n_i-1} \geq \alpha_i^{n_i} \) is not true, the assumption is wrong and \( n_i \) must be already beyond the desired truncation point \( (n_i > z_i) \). Thus the truncation point is chosen as \( z_i = n_i - 1 \) in SBRA-ideal. In general, SBRA-ideal uses (4.7) to approximate (4.1).

\[
A_i^{z_i-1} = \frac{D_i^{z_i-1}}{R_i^{z_i}} - R_i^{z_i-1} \geq \alpha_i^{z_i}
\] (4.7)

In the final non-ideal form of SBRA which we will simply call SBRA, \( \alpha_i^{z_i} \) will be approximated by a quantity which can be computed in a causal manner. SBRA also chooses \( z_i = n_i \) instead of \( z_i = n_i - 1 \) in step 5. An advantage of this choice is that the encoding of the last \( n_i^{th} \) coding pass in step 3 before proceeding to the next code block is not wasted. In our experiments, the choice of \( z_i = n_i \) gives similar rate-distortion performance as \( z_i = n_i - 1 \).

4.2.2 Determination of \( \alpha \)

We now describe our choices of \( \alpha_i^{z_i} \). As \( \lambda_{\text{optimal}} \) is unknown, we design a casual \( \alpha_i^{z_i} \) by enforcing the rate constraint as

\[
R = \sum_k R_k^{z_i} = R_{\text{budget}}
\] (4.8)

Let \( B_i \) be the code-block currently being encoded. We define \( B_{\text{encoded}} = \{ B_j \}_{j \in i} \) which is the set of previously encoded code-blocks and \( B_{\text{encoded}} = \{ B_j \}_{j \neq i} \) which is the set of
future code-blocks. For the current code-block $B_i$, suppose the feasible truncation points $1, \ldots, n_i$ have already been encoded and $n_i < z_i$. We define $D_{\text{remain}}^{(l,n_i)}$ and $R_{\text{remain}}^{(l,n_i)}$ as

$$D_{\text{remain}}^{(l,n_i)} = D_i^{n_i} + \sum_{k \in B_{\text{encoded}}} D_k^{0}$$  \hspace{1cm} (4.9)

and

$$R_{\text{remain}}^{(l,n_i)} = R_{\text{budget}} - R_i^{n_i} - \sum_{k \in B_{\text{encoded}}} R_k^{z_k}$$ \hspace{1cm} (4.10)

$$= R_i^{z_i} - R_i^{n_i} + \sum_{k \in B_{\text{encoded}}} R_k^{z_k}$$ \hspace{1cm} (4.11)

where $D_k^{0}$ is the distortion incurred by reconstructing zero (null) data for code-blocks $B_k$. Note that (4.9) and (4.10) allow both $D_{\text{remain}}^{(l,n_i)}$ and $R_{\text{remain}}^{(l,n_i)}$ to be computed casually.

We assume the remaining bit budget, $R_{\text{remain}}^{(l,n_i)}$, for the unencoded data is greater than zero because the rate control needs to continue only when $R_{\text{remain}}^{(l,n_i)} > 0$. As $n_i < z_i$ and assuming $0 < z_k$ for future $k > i$, we want to find a casual $\alpha_i^{n_i}$ such that (4.7) is true for $B_i$ and the subsequent blocks $\{B_j\}_{j>i}$. Combining (4.6) and (4.7), we have

$$\left\{ \begin{array}{l}
D_i^{n_i} \geq \alpha_i^{z_i} (R_i^{z_i} - R_i^{n_i}) \\
D_k^{0} \geq \alpha_k^{z_k} (R_k^{z_k} - R_k^{0}) \\
\forall k \in B_{\text{encoded}}
\end{array} \right.$$ \hspace{1cm} (4.12)

where $R_k^{0}$ is the rate that the bit-stream is totally truncated and thus $R_k^{0} = 0$. Summing over the set of inequality in (4.12), we have

$$D_i^{n_i} + \sum_{k \in B_{\text{encoded}}} D_k^{0} \geq \alpha_i^{z_i} (R_i^{z_i} - R_i^{n_i}) + \sum_{k \in B_{\text{encoded}}} \alpha_k^{z_k} R_k^{z_k}$$ \hspace{1cm} (4.13)
Now we make the simplifying assumption that $\alpha_{i+k}^n = \alpha_i^i$ for all $k > i$. As we assume $n_i = z_i$ in step 4 of SBRA-ideal, we can change $\alpha_i^i$ to become $\alpha_i^n$. Then, (4.13) becomes

$$D_i^n + \sum_{k \in B_{\text{seced}}} D_k^i \geq \alpha_i^n (R_i^n - R_i^n + \sum_{k \in B_{\text{seced}}} R_k^n)$$  \hspace{1cm} (4.14)

Substituting (4.9) and (4.11) into (4.14), we have $D_i^{(i,n)}(\text{remain}) \geq \alpha_i^n R_i^{(i,n)}(\text{remain})$, or

$$\alpha_i^n \leq \frac{D_i^{(i,n)}(\text{remain})}{R_i^{(i,n)}(\text{remain})}$$  \hspace{1cm} (4.15)

As $D_i^{(i,n)}(\text{remain}) / R_i^{(i,n)}(\text{remain})$ is a non-negative number, this implies that

$$\alpha_i^n = \frac{D_i^{(i,n)}(\text{remain})}{R_i^{(i,n)}(\text{remain})} \cdot \frac{1}{\beta_i^n}$$  \hspace{1cm} (4.16)

where $\beta_i^n \geq 1$. This expression of $\alpha_i^n$ will be used in our proposed SBRA algorithm instead of that in step 4 of SBRA-ideal. We will use two choices of weighting factors $\beta_i^n$ to achieve different complexity-performance trade-off and will be discussed later. Substituting (4.16) into (4.7), we have

$$\frac{D_i^{(i,n)-1}}{R_i^n - R_i^{i-1}} \geq \frac{D_i^{(i,n)}(\text{remain})}{R_i^{(i,n)}(\text{remain})} \cdot \frac{1}{\beta_i^n}$$  \hspace{1cm} (4.17)

Or, equivalently, the condition in step 5 of SBRA-ideal or (4.7) can be replaced by

$$R_i^n - R_i^{i-1} \leq \frac{T_i^n}{\beta_i^n} \cdot \frac{D_i^{(i,n)}(\text{remain})}{D_i^{(i,n)}}$$  \hspace{1cm} (4.18)

The corresponding stopping criterion is $R_i^n - R_i^{i-1} > T_i^n$. If $\beta_i^n$ is available, we note that the condition in (4.18) is in casual form as all the variables can be computed without encoding the future code-blocks. Thus we can select the truncation points in parallel with tier-1 coding using (4.18).
4.2.3 SBRA Formation

In real situations, the R-D ratios \( R_i^{n-1} \) evaluated at coding-pass boundary points (feasible truncation points) may not be always strictly decreasing as can be observed in Fig. 12 which shows the actual R-D ratios of the first code-block of the test image ‘lena’. Thus, instead of evaluating the R-D ratios at all the coding-pass boundary points, we evaluate the R-D ratio (or equivalently the threshold \( T_i^n \)) only at bit-plane boundary points in the proposed SBRA. This is why our algorithm is called successive bit-plane rate allocation (SBRA). There are two major advantages in evaluating the R-D ratio at bit-plane boundary points. Firstly and most importantly, we can compute the distortion without distinguishing which coefficient is encoded by which coding pass in a bit-plane, which simplifies greatly the computation of \( T_i^n \). Secondly, we observe that the R-D ratio evaluated at bit-plane boundaries are more likely to be strictly decreasing.

![Graph showing R-D ratio](image)

Fig. 12: Plot of R-D ratio curve of first code-block for test image ‘lena’. Bigger dots represent bit-plane boundaries.
The complexity to compute $T_i^{n_i}$ is low because the incremental complexity to compute $R_{\text{remain}}^{(i,n_i)}$ and $\beta_i^{n_i}$ is very small and there is a fast method to compute $D_{\text{remain}}^{(i,n_i)}$ and $D_i^n$. In JPEG2000, the distortion can be computed by summing the distortion reduction of each coefficient using a fast look-up table method [1]. To compute the total distortion that is required at the beginning of SBRA, we can simply use the fast look-up table method to pre-compute the distortion reduction of all the bit-planes and add them together. Our choice in SBRA to evaluate the R-D ratio only at bit-plane boundary points greatly simplifies this complexity because the distortion reductions values can be computed in the bit-plane level without performing coefficient bit modeling. The distortion reduction values can then be stored and re-used in computing $D_i^n$. This greatly simplifies the distortion computation.

In JPEG2000, the bit-plane boundaries are at $n_i = 1, 4, 7, 10, \ldots$. So the stopping criterion in (4.18) is modified to become

$$
\begin{align*}
R_i^{n_i} - R_i^0 & \geq T_i^0 \quad \text{for } n_i = 1 \\
R_i^{n_i} - R_i^{n_{i-3}} & \geq T_i^{n_{i-3}} \quad \text{for } n_i = 4, 7, 10, \ldots
\end{align*}
$$

(4.19)

However, in order to have a finer selection of truncation point, instead of using (4.19), we choose the stopping criterion as

$$
R_i^{n_i} - R_i^{b_i(n_i-1)} \geq T_i^{b_i(n_i-1)} \quad \text{for } n_i = 1, 2, 3, \ldots
$$

(4.20)

where

$$
b_i(x) = \begin{cases} 
0 & \text{for } x = 0 \\
3 \cdot \left\lfloor x / 3 \right\rfloor - 2 & \text{otherwise}
\end{cases}
$$

(4.21)
Note that if \( T_i^{b_i (n_{i-1})} \) in (4.20) is small enough, the stopping criteria in (4.20) will always be true. Thus we can terminate early the tier-1 encoding process of a code block when

\[
T_i^{b_i (n_{i-1})} \leq Th
\]  
(4.22)

for some threshold \( Th \). A simple choice of \( Th \) can be zero and it is the choice in our implementation. So, here is the final, non-ideal SBRA algorithm:

1. Initialize \( i = 1 \).
2. Initialize \( n_i = 0 \).
3. Set \( n_i = n_i + 1 \). Assume \( n_i = z_i \) and compute \( T_i^{b_i (n_{i-1})} \).
4. If \( T_i^{b_i (n_{i-1})} \leq Th \), choose \( z_i = n_i - 1 \) as the selected truncation point and go to step 8.
5. Encode the \( n_i^{th} \) coding-pass of code-block \( B_i \) (i.e. encode \( B_i \) up to the feasible truncation point \( n_i \)).
6. Compute \( R_i^{n_i} - R_i^{b_i (n_{i-1})} \).
7. If \( R_i^{n_i} - R_i^{b_i (n_{i-1})} \leq T_i^{b_i (n_{i-1})} \), go to step 3. Else choose \( z_i = n_i \) as the selected truncation point and go to step 8.
8. If \( B_i \) is not the last code-block, set \( i = i + 1 \) and go to step 2.

As SBRA requires only the R-D information of the current code block, the R-D information of the previously encoded code block can be forgotten. SBRA is thus memoryless and casual. And the encoded code-block data can be sent out immediately.
4.2.3.1 Determination of $\beta$

Now we will describe our choices of $\beta_i^n$ to be used in computing $T_i^{h(n-1)}$. Typically the characteristics of wavelet coefficients within the same subband are similar. Thus we choose to use one weighting factor value for each wavelet subband. We denote the single weighting factor for each subband as $\beta_{r,\theta}$ where $r$ is the resolution level and $\theta$ is the orientation, as shown in Fig. 13. Note that subbands at different resolution levels contain different amount of code-blocks. And all the code-blocks within a subband have the same weighting factor with $\beta_i^n = \beta_{r,\theta}$ for all $i$ in the $(r,\theta)$ subband, or $\forall i \in (r,\theta)$. A single index number is also given to each $(r,\theta)$ subband as shown in Fig. 13, with smaller indices for lower subbands.

<table>
<thead>
<tr>
<th>(0,0)</th>
<th>(1,1)</th>
<th>(2,1)</th>
<th>(3,1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{0,0} = 1$</td>
<td>$k_{1,1} = 2$</td>
<td>$k_{2,1} = 5$</td>
<td>$k_{3,1} = 8$</td>
</tr>
<tr>
<td>(1,2)</td>
<td>(1,3)</td>
<td>(2,2)</td>
<td>(2,3)</td>
</tr>
<tr>
<td>$k_{1,2} = 3$</td>
<td>$k_{1,3} = 4$</td>
<td>$k_{2,2} = 6$</td>
<td>$k_{2,3} = 7$</td>
</tr>
<tr>
<td>(3,2)</td>
<td>(3,3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{3,2} = 9$</td>
<td>$k_{3,3} = 10$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 13: Indexing of subbands. Each subband is indexed by a resolution level and an orientation $(r,q)$. The lower single index $k$ is computed by (4.24). This example shows a three level DWT decomposition.
A simple form of SBRA is to ignore the weighting factors by forcing $\beta_i^\circ = 1$. This non-weighted method is called SBRA-1. However the simple SBRA-1 has problems as shown in Fig. 14, which shows the average rate difference between PCRD and SBRA-1 over many test images. The test images of various size and nature are shown in Fig. 15. Fig. 14 suggests that SBRA-1 tends to allocate lower rates than PCRD at lower resolution subbands (with smaller subband index) and the rate allocation difference is approximately linear with the subband index. As the weighting factor in (4.20) affects linearly the threshold for stopping encoding for any code-block, a larger weighting factor can make it more difficult to stop and thus increasing the total bits allocated for the code-block. In other words, we can have a truncation point later than that in SBRA-1. We then assume that the weighting factor is proportional to the achieved bit-rate for a given code-block. Thus we model $\beta_{r,\theta}$ by a linear equation as

$$\beta_{r,\theta} = 1 + m(M - k_{r,\theta})$$ \hspace{1cm} (4.23)

and

$$k_{r,\theta} = \begin{cases} 1 & \text{for } (r, \theta) = (0,0) \\ 3(r-1) + \theta + 1 & \text{otherwise} \end{cases}$$ \hspace{1cm} (4.24)

where $m$ is the slope of the linear curve. $M$ is the total number of subbands and $(r, \theta)$ is indexed as shown in Fig. 13. We observe, in Fig. 14, that the rate allocation difference can be approximated by a linear relationship with the target bit-rate. Thus we simply model $m$ as

$$m = B_{\text{budget}} / \Delta$$ \hspace{1cm} (4.25)

where $\Delta$ is a pre-defined constant and $B_{\text{budget}}$ is the target bit-rate in term of bits per pixel (bpp).
Fig. 14: Average rate difference between PCRD and SBRA-I (SBRA-I–PCRD) for sixteen subbands at different target bit-rates.
Here we assume that the linear model is applied for the coding order as shown in Fig. 16. We denote the proposed SBRA method using this linear model as SBRA-w. As we use five-level DWT decomposition in all our simulations/implementation, we will suggest a value of $\Delta$ based on that. Fig. 17 shows the relationship between the values of $\Delta$ and average PSNR performance of SBRA-w using the test images in Fig. 15. It shows that the improvement of the PSNR performance starts to saturate after the values of three and the value of five gives a relatively good result among them. Thus we set $\Delta = 5$ in
our implementation. Fig. 18 shows the average rate difference between PCRD and SBRA-w using $\Delta = 5$. It shows that SBRA-w can highly reduce the rate allocation difference when compared with SBRA-1.

Fig. 16: Scan order of (a) subband and (b) code-block within a subband.

Fig. 17: Average PSNR difference between PCRD and SBRA-w (PCRD–SBRA-w) for different values of $\Delta$ at different target bit-rates.

39
Fig. 18: Average rate difference between PCRD and SBRA-w (SBRA-w–PCRD) for sixteen subbands at different target bit-rates.

4.3 Priority Scanning Rate Allocation (PSRA)

The second proposed fast rate control method is called priority scanning rate allocation (PSRA). By examining the PCRD scheme, we observe that, if we sort the R-D slope of all the feasible truncation point \( n_i \) in descending order and keep the order in a sorted list, the optimal rate allocation can also be achieved by selecting the truncation point from the top of the sorted list according to the sorted order. This implies that, once we know the R-D slope order, we do not even need to know the actual value of the R-D slope. The problem now becomes how to obtain the order without knowing all the values of the R-D slope.
However, it is unlikely that we can obtain the actual order list without encoding all the feasible coding-passes to obtain the actual R-D slope. Instead, we are going to predict the order based on some assumptions. Based on the assumed strictly decreasing property of R-D slope function, an earlier truncation point \( i \) will have larger R-D slope than later truncation point \( j \) (\( j > i \)) within a code-block. We further assume that a bit-plane level \( k \) of any code-block has a R-D slope larger than that in bit-plane levels lower than \( k \) of other code-blocks.

With these assumptions, we set the coding priority (order) based on the bit-plane level and coding pass type within an image. The coding pass with higher priority will be tier-1 encoded first followed by coding passes with lower priority. The proposed method is called priority scanning rate allocation (PSRA). The detail PSRA is as follows:

1. (Initialization) For the \( i^{th} \) code block \( B_i \), the initial code-block priority is equal to the maximum number of coding passes required to fully encode the data in the code-block and is calculated by

\[
P_i = 3 * j_i^0 - 2
\]  \hspace{1cm} (4.26)

where \( j_i^0 \) is the maximum number of bit-plane level required to fully encode the data in the code-block. The current coding priority is defined as

\[
P_c = \max_i (P_i)
\]  \hspace{1cm} (4.27)

2. (Priority Scanning) Visit each code-block according to the scan order as shown in Fig. 16. For each code-block \( B_i \), condition C1 is checked followed by either C2 or C3.
C1: If the code-block priority is equal to the current coding priority \((P_i = P_e)\), encode the first unencoded coding pass of \(B_i\), reduce \(P_i\) by one, calculate the R-D slope by (3.5) and check the condition C2. Else check the condition C3.

C2: If the accumulated bit-rate is larger than the target bit-rate, find and discard the encoded coding pass(es) which has the minimum R-D slope(s) such that the accumulated bit-rate is less than or equal to the target bit-rate. The whole tier-1 coding process will be terminated at this point. Else check the condition C3.

C3: If the code-block is the last one in the scan order, reduce \(P_e\) by one and the next code-block will be the first one in the scan order. Else visit the following code block.

The code-block priority \(P_i\) in (4.26) keeps track of the current priority level of individual code-blocks. The current coding priority \(P_e\) in (4.27) keeps track of the truncation point level to be processed in the current round within an image. In step C1, code-blocks with \(P_i\) less than \(P_e\) would not be encoded. In step C2, the removal of the encoded coding pass with minimum R-D slope helps to achieve high PSNR.

As shown in our experimental results, PSRA can have good PSNR performance for most test images. Unlike SBRA, the encoded code-block data in PSRA can be sent out only after the whole PSRA process is finished.
4.4 Priority Scanning with Optimal Truncation (PSOT)

The PCRD method achieves the best rate-distortion performance but is not efficient because it requires encoding all the available data. Compared with PCRD, PSRA encodes a significantly smaller amount of data but the rate-distortion performance is less than optimal. Here we propose a new method called priority scanning with optimal truncation (PSOT) to achieve the same optimal rate-distortion performance as PCRD with computational complexity less than PCRD.

We first use the PSRA method to estimate a R-D slope that could be close to the optimal $\lambda$. Then a minimum slope rejection method is used to terminate the tier-1 coding process of each code block such that the last calculated R-D slope of each code block is less than or equal to the estimated R-D slope. Finally, PCRD is applied to the encoded data. By doing so, instead of encoding all available data, only a smaller amount of data is required to be encoded. This proposed scheme is performed as follows:

1. (Initialization & Priority Scanning) Perform PSRA as in Section IV. However, instead of termination, if the accumulated bit rate is larger than the target bit rate, go to step 2.

2. (Minimum Slope Rejection) Find the minimum slope value $S_{min}$ among the set of calculated R-D slopes in step 1. For each code-block, continue encoding as long as the R-D slope is greater than $S_{min}$. Once all the code-blocks have terminated, go to step 3.

3. (Optimal Truncation) This step is essentially the PCRD method. The $\lambda_{optimal}$ is found among the encoded data. Then the coding passes with R-D slopes greater than or equal to the $\lambda_{optimal}$ are included in the final code-stream.

43
In Step 2, the $S_{\min}$ is computed when the accumulated bit rate is larger than the target bit rate. As a result, $S_{\min}$ should be less than $\lambda_{\text{optimal}}$ because, as mentioned before, $\lambda_{\text{optimal}}$ is the minimum value of $\lambda$ in (3.6) which satisfies the rate constraint in (3.3).

With $S_{\min} < \lambda_{\text{optimal}}$, we have

$$z_j(\hat{\lambda}) = \max\{j \in N_+ | S_j^l \geq \hat{\lambda}\} \leq \max\{j \in N_+ | S_j^l \geq S_{\min}\} \quad (4.28)$$

We only need to encode all the feasible truncation points up to $S_{\min}$ in Step 3 and would be guaranteed to have included all the feasible truncation points that would be selected by PCRD into the code-stream. Thus, PCRD can be applied in Step 4 and optimal rate allocation is achieved. The major advantage of PSOT over PCRD is that the redundant computation and redundant memory can be greatly reduced. Similar to PSRA, the encoding of the feasible truncation points until the R-D slope is less than $S_{\min}$ has to be finished for all the code-blocks in Step 3 before the encoded code-stream can be generated in Step 4.

4.5 Simulation Results

The proposed rate control methods were tested on seven 512x512 popular test images including “lena”, “barbara”, “goldhill”, “boat”, “mandrill”, “peppers” and “zelda” as well as the eight test images from the JPEG2000 test suite, “aerial2”, “bike”, “café”, “chart”, “mat”, “target”, “tools” and “woman”. The preview of those images is shown in Fig. 15. The proposed scheme was implemented in the reference software called “Jasper” [27, 28] which is defined in Part 5 of the JPEG2000 standard [26]. In all
images, we used the Daubechies 9/7 bi-orthogonal wavelet filters with five-level DWT decomposition (or six resolution levels) and the code-block size was 64x64.

Here we first use a powerful profiling tool called Vtune\(^1\) to check the computational complexity of different JPEG2000 components. The computational complexity is measured in term of number of CPU clock cycle which is the smallest time unit in a computer (PC). Fig. 19 shows the computational complexity profiles of JPEG2000 using PCRD in four test images, and Fig. 20 shows the average complexity profile of the fifteen test images in Fig. 15.

---

Fig. 19: Computational complexity profiles of JPEG2000 using PCRD in test images (a) 'barbara', (b) 'aerial2', (c) 'bike' and (d) 'tools'.

\(^1\) Intel\textregistered Vtune\textsuperscript{TM} Performance Analyzer 6.1
Fig. 20: Average computational complexity profile of JPEG2000 using PCRD in the fifteen test images in Fig. 15.

We can see that, when using PCRD, the tier-1 encoding can be the largest potion of computational complexity in JPEG2000. The DWT is implemented by the fast lifting-based filtering and it is, in average, the second most computational intensive part. The input/output (IO) and memory initialization is about 16% of the total complexity in average. The other parts include quantization, packetization and some overhead processes consume about 24% of the total encoding time.

After profiling the total encoding process, we then profile the computational complexity and memory usage within the tier-1 encoding process only. Some typical computational complexity results are shown in Table 4 for some test images. The computational complexity is measured in term of number of CPU clock cycles per pixel required for tier-1 coding. The SBRA-w in the forth column of Table 4 represents the SBRA method using the linear rate model with Δ equal to five. The computational
complexity of Tier-1 coding using SBRA-1, SBRA-w, and PSRA are similar and can be up to 90% less than PCRD. The PSOT requires more computation than SBRA-1, SBRA-w and PSRA but can still save up to 80% of computation when compared with PCRD. Fig. 21 shows the average computational complexity results over all the test images. SBRA-1, SBRA-2 and PSRA can save more than 94% and 85% of the tier-1 coding complexity at the bit-rate of 0.0625bpp and 0.025bpp respectively, while PSOT can save more than 85% and 65% respectively.

<table>
<thead>
<tr>
<th>Lena (512x512)</th>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>1184</td>
<td>76</td>
<td>78</td>
<td>75</td>
<td>163</td>
<td></td>
</tr>
<tr>
<td>0.125</td>
<td>1184</td>
<td>116</td>
<td>115</td>
<td>112</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>1184</td>
<td>203</td>
<td>197</td>
<td>192</td>
<td>420</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1184</td>
<td>358</td>
<td>347</td>
<td>343</td>
<td>570</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1184</td>
<td>567</td>
<td>557</td>
<td>566</td>
<td>803</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mandrill (512x512)</th>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>1803</td>
<td>71</td>
<td>70</td>
<td>75</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td>0.125</td>
<td>1803</td>
<td>99</td>
<td>99</td>
<td>120</td>
<td>262</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>1803</td>
<td>166</td>
<td>161</td>
<td>168</td>
<td>365</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1803</td>
<td>276</td>
<td>265</td>
<td>285</td>
<td>480</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1803</td>
<td>466</td>
<td>460</td>
<td>459</td>
<td>722</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aerial2 (2048x2048)</th>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>1581</td>
<td>74</td>
<td>72</td>
<td>70</td>
<td>210</td>
<td></td>
</tr>
<tr>
<td>0.125</td>
<td>1581</td>
<td>109</td>
<td>107</td>
<td>106</td>
<td>316</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>1581</td>
<td>168</td>
<td>166</td>
<td>165</td>
<td>541</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1581</td>
<td>274</td>
<td>267</td>
<td>300</td>
<td>615</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1581</td>
<td>465</td>
<td>455</td>
<td>466</td>
<td>979</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Café (2560x2048)</th>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>1686</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>179</td>
<td></td>
</tr>
<tr>
<td>0.125</td>
<td>1686</td>
<td>110</td>
<td>107</td>
<td>106</td>
<td>373</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>1686</td>
<td>176</td>
<td>173</td>
<td>182</td>
<td>604</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1686</td>
<td>299</td>
<td>297</td>
<td>291</td>
<td>605</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1686</td>
<td>544</td>
<td>537</td>
<td>521</td>
<td>1175</td>
<td></td>
</tr>
<tr>
<td>Bit-rate</td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>0.0625</td>
<td>1056</td>
<td>82</td>
<td>81</td>
<td>76</td>
<td>264</td>
<td></td>
</tr>
<tr>
<td>0.125</td>
<td>1056</td>
<td>127</td>
<td>127</td>
<td>122</td>
<td>311</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>1056</td>
<td>217</td>
<td>216</td>
<td>212</td>
<td>518</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1056</td>
<td>380</td>
<td>377</td>
<td>370</td>
<td>784</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1056</td>
<td>635</td>
<td>632</td>
<td>630</td>
<td>1035</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>1725</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>190</td>
</tr>
<tr>
<td>0.125</td>
<td>1725</td>
<td>123</td>
<td>121</td>
<td>120</td>
<td>313</td>
</tr>
<tr>
<td>0.25</td>
<td>1725</td>
<td>202</td>
<td>198</td>
<td>203</td>
<td>466</td>
</tr>
<tr>
<td>0.5</td>
<td>1725</td>
<td>347</td>
<td>338</td>
<td>344</td>
<td>633</td>
</tr>
<tr>
<td>1.0</td>
<td>1725</td>
<td>605</td>
<td>602</td>
<td>594</td>
<td>909</td>
</tr>
</tbody>
</table>

Table 4: Computational complexity results, measured in number of CPU clock cycles per pixel required for tier-1 encoding. (Profile at PIII-800 384M RAM, Intel Vtune performance analyzer 6.1 and Windows XP).

![Chart](chart.png)

Fig. 21: Average number of CPU clock cycles per pixel required for tier-1 coding for the test images in Fig. 15.
The SBRA method is a memoryless and causal rate control scheme. The codeblock bit-stream can be sent out immediately after the determination of the truncation point. Thus the working memory size required for storing the code-block bit-stream is approximately zero. Table 5 provides the memory requirement results in term of bits per pixel. The memory requirement of the PSRA method is approximately equal to the target bit-rate. As the PSOT method needs to encode more data to find the optimal truncation points, the memory requirement is about two or three times more than that of the PSRA. The average working memory size results are shown in Fig. 22. PSRA and PSOT require about 92% and 75% less working memory size than PCRD respectively at the bit-rate of 0.25bpp. As mentioned before, SBRA requires approximately zero working memory size.

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>Lena (512x512)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
</tr>
<tr>
<td>0.0625</td>
<td>2.749</td>
<td>0</td>
<td>0</td>
<td>0.056</td>
<td>0.161</td>
</tr>
<tr>
<td>0.125</td>
<td>2.749</td>
<td>0</td>
<td>0</td>
<td>0.118</td>
<td>0.197</td>
</tr>
<tr>
<td>0.25</td>
<td>2.749</td>
<td>0</td>
<td>0</td>
<td>0.241</td>
<td>0.622</td>
</tr>
<tr>
<td>0.5</td>
<td>2.749</td>
<td>0</td>
<td>0</td>
<td>0.488</td>
<td>1.011</td>
</tr>
<tr>
<td>1.0</td>
<td>2.749</td>
<td>0</td>
<td>0</td>
<td>0.987</td>
<td>1.605</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>Mandrill (512x512)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
</tr>
<tr>
<td>0.0625</td>
<td>4.877</td>
<td>0</td>
<td>0</td>
<td>0.056</td>
<td>0.217</td>
</tr>
<tr>
<td>0.125</td>
<td>4.877</td>
<td>0</td>
<td>0</td>
<td>0.120</td>
<td>0.457</td>
</tr>
<tr>
<td>0.25</td>
<td>4.877</td>
<td>0</td>
<td>0</td>
<td>0.241</td>
<td>0.706</td>
</tr>
<tr>
<td>0.5</td>
<td>4.877</td>
<td>0</td>
<td>0</td>
<td>0.493</td>
<td>1.040</td>
</tr>
<tr>
<td>1.0</td>
<td>4.877</td>
<td>0</td>
<td>0</td>
<td>0.988</td>
<td>1.679</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>Aerial2 (2048x2048)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
</tr>
<tr>
<td>0.0625</td>
<td>4.173</td>
<td>0</td>
<td>0</td>
<td>0.062</td>
<td>0.301</td>
</tr>
<tr>
<td>0.125</td>
<td>4.173</td>
<td>0</td>
<td>0</td>
<td>0.124</td>
<td>0.561</td>
</tr>
<tr>
<td>0.25</td>
<td>4.173</td>
<td>0</td>
<td>0</td>
<td>0.248</td>
<td>1.215</td>
</tr>
<tr>
<td>0.5</td>
<td>4.173</td>
<td>0</td>
<td>0</td>
<td>0.496</td>
<td>1.366</td>
</tr>
<tr>
<td>1.0</td>
<td>4.173</td>
<td>0</td>
<td>0</td>
<td>0.993</td>
<td>2.539</td>
</tr>
</tbody>
</table>

49
### Table 5: Working memory size results, measured in bits per pixel.

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>3.975</td>
<td>0</td>
<td>0</td>
<td>0.061</td>
<td>0.231</td>
</tr>
<tr>
<td>0.125</td>
<td>3.975</td>
<td>0</td>
<td>0</td>
<td>0.123</td>
<td>0.622</td>
</tr>
<tr>
<td>0.25</td>
<td>3.975</td>
<td>0</td>
<td>0</td>
<td>0.246</td>
<td>1.134</td>
</tr>
<tr>
<td>0.5</td>
<td>3.975</td>
<td>0</td>
<td>0</td>
<td>0.496</td>
<td>1.134</td>
</tr>
<tr>
<td>1.0</td>
<td>3.975</td>
<td>0</td>
<td>0</td>
<td>0.994</td>
<td>2.600</td>
</tr>
</tbody>
</table>

### Chart (2347x1688)

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>1.914</td>
<td>0</td>
<td>0</td>
<td>0.061</td>
<td>0.316</td>
</tr>
<tr>
<td>0.125</td>
<td>1.914</td>
<td>0</td>
<td>0</td>
<td>0.123</td>
<td>0.394</td>
</tr>
<tr>
<td>0.25</td>
<td>1.914</td>
<td>0</td>
<td>0</td>
<td>0.245</td>
<td>0.758</td>
</tr>
<tr>
<td>0.5</td>
<td>1.914</td>
<td>0</td>
<td>0</td>
<td>0.495</td>
<td>1.331</td>
</tr>
<tr>
<td>1.0</td>
<td>1.914</td>
<td>0</td>
<td>0</td>
<td>0.989</td>
<td>1.887</td>
</tr>
</tbody>
</table>

### Tools (1200x1524)

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>4.114</td>
<td>0</td>
<td>0</td>
<td>0.060</td>
<td>0.219</td>
</tr>
<tr>
<td>0.125</td>
<td>4.114</td>
<td>0</td>
<td>0</td>
<td>0.122</td>
<td>0.458</td>
</tr>
<tr>
<td>0.25</td>
<td>4.114</td>
<td>0</td>
<td>0</td>
<td>0.245</td>
<td>0.754</td>
</tr>
<tr>
<td>0.5</td>
<td>4.114</td>
<td>0</td>
<td>0</td>
<td>0.494</td>
<td>1.093</td>
</tr>
<tr>
<td>1.0</td>
<td>4.114</td>
<td>0</td>
<td>0</td>
<td>0.991</td>
<td>1.768</td>
</tr>
</tbody>
</table>

Fig. 22: Average working memory size for the test images in Fig. 15.
Table 6 shows the PSNR results of six test images and Fig. 23 shows the average PSNR difference between the proposed methods and the PCRD. PSOT achieves the optimal PSNR while PSRA has slightly lower PSNR than PCRD. SBRA-w has considerably higher PSNR than SBRA-1 especially at high bit-rate. Although SBRA has the lowest PSNR, the actually visual quality degradation between SBRA and PCRD is very small. Fig. 24-26 shows the visual quality of ‘lena’ using PCRD, SBRA-1, SBRA-w and PSRA at the bit-rate of 0.125bpp, 0.25bpp and 1bpp respectively. We do not show the visual quality of PSOT because it is the same as PCRD. For all the proposed methods, the visual quality is not noticeable when compared with PCRD.

<table>
<thead>
<tr>
<th>Lena (512x512)</th>
<th>Bit-rate</th>
<th>PCRD</th>
<th>SBRA-1</th>
<th>SBRA-w</th>
<th>PSRA</th>
<th>PSOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0625</td>
<td>27.95</td>
<td>27.94</td>
<td>27.94</td>
<td>27.94</td>
<td>27.94</td>
<td>29.95</td>
</tr>
<tr>
<td>0.125</td>
<td>30.98</td>
<td>30.84</td>
<td>30.85</td>
<td>30.90</td>
<td>30.98</td>
<td>30.98</td>
</tr>
<tr>
<td>0.25</td>
<td>34.11</td>
<td>33.78</td>
<td>33.87</td>
<td>34.03</td>
<td>34.11</td>
<td>34.11</td>
</tr>
<tr>
<td>0.5</td>
<td>37.29</td>
<td>36.66</td>
<td>36.95</td>
<td>37.23</td>
<td>37.29</td>
<td>37.29</td>
</tr>
<tr>
<td>1.0</td>
<td>40.38</td>
<td>39.38</td>
<td>39.82</td>
<td>40.34</td>
<td>40.38</td>
<td>40.38</td>
</tr>
<tr>
<td>Mandrill (512x512)</td>
<td>Bit-rate</td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
</tr>
<tr>
<td>0.0625</td>
<td>20.65</td>
<td>20.44</td>
<td>20.45</td>
<td>20.59</td>
<td>20.65</td>
<td>20.65</td>
</tr>
<tr>
<td>0.25</td>
<td>23.18</td>
<td>22.65</td>
<td>22.61</td>
<td>23.14</td>
<td>23.18</td>
<td>23.18</td>
</tr>
<tr>
<td>0.5</td>
<td>25.54</td>
<td>25.13</td>
<td>24.48</td>
<td>25.41</td>
<td>25.54</td>
<td>25.54</td>
</tr>
<tr>
<td>1.0</td>
<td>29.07</td>
<td>28.80</td>
<td>28.73</td>
<td>29.02</td>
<td>29.07</td>
<td>29.07</td>
</tr>
<tr>
<td>Aerial2 (2048x2048)</td>
<td>Bit-rate</td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
</tr>
<tr>
<td>0.0625</td>
<td>24.61</td>
<td>24.19</td>
<td>24.19</td>
<td>24.55</td>
<td>24.61</td>
<td>24.61</td>
</tr>
<tr>
<td>0.25</td>
<td>28.57</td>
<td>28.20</td>
<td>28.28</td>
<td>28.51</td>
<td>28.57</td>
<td>28.57</td>
</tr>
<tr>
<td>0.5</td>
<td>30.62</td>
<td>30.18</td>
<td>30.29</td>
<td>30.49</td>
<td>30.62</td>
<td>30.62</td>
</tr>
<tr>
<td>1.0</td>
<td>33.26</td>
<td>32.84</td>
<td>32.92</td>
<td>33.20</td>
<td>33.26</td>
<td>33.26</td>
</tr>
<tr>
<td>Café (2560x2048)</td>
<td>Bit-rate</td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
</tr>
<tr>
<td>0.0625</td>
<td>19.03</td>
<td>18.86</td>
<td>18.87</td>
<td>18.96</td>
<td>19.03</td>
<td>19.03</td>
</tr>
<tr>
<td>0.125</td>
<td>20.74</td>
<td>20.51</td>
<td>20.55</td>
<td>20.67</td>
<td>20.74</td>
<td>20.74</td>
</tr>
<tr>
<td>Bit-rate</td>
<td>PCRD</td>
<td>SBRA-1</td>
<td>SBRA-w</td>
<td>PSRA</td>
<td>PSOT</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>0.0625</td>
<td>25.86</td>
<td>25.61</td>
<td>25.64</td>
<td>25.61</td>
<td>25.86</td>
<td></td>
</tr>
<tr>
<td>0.125</td>
<td>28.75</td>
<td>28.07</td>
<td>28.25</td>
<td>28.44</td>
<td>28.75</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>32.45</td>
<td>31.44</td>
<td>31.54</td>
<td>32.23</td>
<td>32.45</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>37.44</td>
<td>35.81</td>
<td>36.73</td>
<td>37.19</td>
<td>37.44</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>43.69</td>
<td>42.83</td>
<td>42.89</td>
<td>43.58</td>
<td>43.69</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: PSNR results, measured in dB.

Fig. 23: Average PSNR difference between proposed methods and PCRD (proposed-PCRD) for the test images in Fig. 15.
Fig. 24: Visual quality of 'lena' compressed at 0.125bpp using (a) PCRD (30.98dB), (b) SBRA-1 (30.84dB), (c) SBRA-w (30.85dB) and (d) PSRA (30.90dB).
Fig. 25: Visual quality of ‘lena’ compressed at 0.25bpp using (a) PCRD (34.11dB), (b) SBRA-1 (33.78dB), (c) SBRA-w (33.87dB) and (d) PSRA (34.03dB).
Fig. 26: Visual quality of 'lena' compressed at 1bpp using (a) PCRD (40.38dB), (b) SBRA-1 (39.38dB), (c) SBRA-w (39.82dB) and (d) PSRA (40.34dB).
Fig. 27 shows the average rate control accuracy for the test images in Fig. 15. It shows that all the proposed methods can achieve the bit-rate very close to the target bit-rate. In average, the rate control error of all the methods is almost negligible.

![Graph](image)

Fig. 27: Average rate control accuracy for the test images in Fig. 15.

### 4.6 Conclusions

In this chapter, three efficient rate control methods, namely SBRA, PSRA, and PSOT, have been proposed for JPEG2000 image coding. The optimal PCRD rate control method has the problems that it requires high computational complexity and large memory storage. The simulation results show that the three proposed rate control methods can highly reduce the computational complexity and working memory size while retaining a good image quality.
The simulation results also suggest that the three proposed methods can give different trade-off among computational complexity, delay and working memory size. The memoryless and causal SBRA can send out the encoded data without proceeding to next code-block. Thus it provides the shortest coding/transmission delay, lowest computational complexity and smallest memory requirement. PSRA can achieve higher PSNR than the SBRA and similar complexity to SBRA. However, the encoded data has to wait until the end of the rate control before sending out. Therefore longer delay and larger memory is required for PSRA. PSOT finds a smaller set of data for optimal truncation and it can achieve optimal PSNR as PCRD. The trade-off is that it requires even higher complexity and working memory size than PSRA.

The proposed three methods provide different features such as coding/transmission delay, computational complexity, memory requirement and PSNR performance. The most suitable method is depended on the application requirements. For a real-time or low delay application or implementation, the SBRA method would be the most suitable method. However, if the PSNR performance is the main issue, the PSOT method may be a better choice.
Chapter 5

Conclusions

This thesis work studied both the optimal and non-optimal rate control methods for JPEG2000 image coding. In this chapter, we summarize the contributions of this thesis work, and propose some research directions for future investigation.

5.1 Contributions of This Thesis

The contributions of the thesis work are listed in the following:

1. Analyzed two kinds of optimal rate control methods including PCRD and LRA.
   a. The SBRA framework provided a close-form equation for fast and causal rate control. A linear model for the weighting factor of SBRA was introduced to improve the performance of SBRA.
   b. The PSRA gave a R-D slope magnitude prediction framework for fast and non-causal rate control.
   c. The PSOT provided a framework in finding a smaller data set for fast optimal rate control.
5.2 Future Work

Some suggestions for future work are given below:

1. The linear weighting factor model of SBRA can improve the PSNR performance. A better model can further improve the performance. One possible method is to model the weighting factor using the subband coefficients statistics.

2. The PSRA uses the bit-plane and coding pass location in predicting the R-D slope magnitude. By introducing more factors, the PSRA performance may be improved.

3. A perceptual model can be integrated into the proposed rate control methods in order to improve the visual quality. By using a perceptual model, we can emphasize a local region by giving more bits to that region. For SBRA, we can make use of the weight factor to achieve the local emphasis. While in PSRA or PSOT, we can use bit-plane shift to achieve that.

4. In hardware implementation of DWT, one possible method is called line-based DWT in which the image is transformed line by line. So the tier-1 coding can be performed once a row of code-blocks is available without waiting entire image to be transformed. In this situation, only the SBRA can be used because it is a memoryless and causal rate control method. A different scanning order and weighting model should be investigated.
References


63


List of Publications


