Efficient and Perceptual Picture Coding Techniques

WEI Zhenyu

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Electronic Engineering

May 2009

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Dedication

To my parents and Lu

Acknowledgment

This thesis would not have been possible without the support from professors, colleagues, friends, and family. It is a pleasure to convey my gratitude to them all in my humble acknowledgment.

In the first place I would like to record my gratitude to my thesis advisor Prof. King Ngi Ngan for his supervision, advice, and guidance from every stage of this research as well as giving me extraordinary experience throughout the work. Above all and the most needed, he provided me unflinching encouragement and support in various ways. I am indebted to him more than he knows.

I would like to thank the other three professors in Image and Video Processing (IVP) Lab, Prof. Wai Kuan Cham, Prof. Thierry Blu and Prof. Hung Tat Tsui for their inspiring comments and instructions. Thanks also go to Dr. Feng Wu, for his mentoring and guidance when I was an intern at Microsoft Research Asia.

I also express my sincere thanks to my colleagues and friends in IVP Lab, Dr. Zhenzhong Chen, Dr. Jian Yao, Dr. Yifeng Jiang, Dr. Zhijun Zhang, Dr. Yu Liu, Jie Dong, Dr. Chun Man Mak, Dr. Hongliang Li, Dr. Wenxian Yang, Dr. Xin Jin, Dr. Haiyan Shu, Dr. Dongdong Zhang, Jie Li, Deqing Sun, Chunhui Cui, Songnan Li, Qiang liu, Wanli Ouyang, Renqi Zhang, Qian Zhang, Kai Lam Tang, Wei Zhang and Cong Zhao. I would like to thank our lab technician Yuk Chung Wong, for his selfless help with computer maintenance.

Last but not least, my sincere gratitude goes to my parents for all their tireless efforts in bringing me up and their never ending love and support. To my dear sister and brother-in-law, thank you for all your support and encouragement. To my lovely new-born nephew, thank you for bringing me happiness. Words fail me to express my appreciation to my wife Lu Yu whose dedication, love and persistent confidence in me, has taken the load off my shoulder.

Abstract

The objective of this thesis is to develop some efficient and perceptual image and video coding techniques. Two parts of the work are investigated in this thesis.

In the first part, some efficient algorithms are proposed to reduce the complexity of H.264 encoder, which is the latest state-of-the-art video coding standard. Intra and Inter mode decision play a vital role in H.264 encoder and can reduce the spatial and temporal redundancy significantly, but the computational cost is also high. Here, a fast Intra mode decision algorithm and a fast Inter mode decision algorithm are proposed. Experimental results show that the proposed algorithms not only save a lot of computational cost, but also maintain coding performance quite well. Moreover, a real time H.264 baseline codec is implemented on mobile device. Based on our real time H.264 codec, an H.264 based mobile video conferencing system is achieved.

The second part of this thesis investigates two kinds of perceptual picture coding techniques. One is the just noticeable distortion (JND) based picture coding. Firstly, a DCT based spatio-temporal JND model is proposed, which is an efficient model to represent the perceptual redundancies existing in images and is consistent with the human visual system (HVS) characteristic. Secondly, the proposed JND model is incorporated into image and video coding to improve the perceptual quality. Based on the JND model, a transparent image coder and a perceptually optimized H.264 video coder are implemented. Another technique is the image compression scheme based on the recent advances in texture synthesis. In this part, an image compression scheme is proposed with the perceptual visual quality as the performance criterion instead of the pixel-wise fidelity. As demonstrated in extensive experiments, the proposed techniques can improve the perceptual quality of picture coding significantly.

摘要

本論文的目標是提出一系列高效且主觀優化的靜止圖像和視頻編碼技術.本論文包括兩部分工作.

本論文的第一部分, 一些高效的算法被提出用以降低目前最新且最先進的視頻編碼標準H.264的算法復雜度. 在H.264算法中, 幀內和幀間模式選擇為編碼性能的提高發揮重要的作用, 空域和時域的冗余被大大降低,但是同時算法復雜度也非常高.本論文提出一種快速幀內模式選擇算法和一種快速幀間模式選擇算法. 實驗證明本文提出的快速算法不僅能够節省大量計算量,而且能够很好的保持編碼性能. 此外, 一個實時的H.264基綫類編解碼器被實現在移動設備上. 基于我們實時的編解碼器,我們實現了一個基于H.264的無綫實時視頻會議系統.

本論文的第二部分探討瞭兩類主觀圖像編碼技術. 一種技術是基于最小可感知失真的圖像編碼技術. 首先,一個基于離散余弦變換的時空域最小可感知失真模型被建立起來, 這是一種描述存在于圖像中主觀冗餘的有效模型,且和人類視覺系統的特性相一致. 然後,本文提出的最小可感知失真模型被應用于靜止圖像和視頻編碼以提高主觀視覺質量. 基于该最小可感知模型,本文提出了一個透明圖像編碼框架,以及一個主觀優化的H.264視頻編解碼器. 另一種技术是基于先進紋理閤成技術的圖像編碼框架.在這一技術中, 主觀視覺質量作爲性能指標,代替了以往像素保真度指標. 大量實驗證明, 本文提出的技術能够明顯提升圖像編碼的主觀質量.

Publications

Journal Papers

- Zhenyu Wei, King N. Ngan and Hongliang Li, "An Efficient Intra Mode Selection Algorithm for H.264 Based on Edge Classification and Rate-Distortion Estimation", *Signal Processing: Image Communication*, Elsevier, Vol.23, No.9, pp.699-710, October 2008.

- Hongliang Li, King N. Ngan and Zhenyu Wei, " Fast and Efficient Method for Block Edge Classification and Its Application in H.264/AVC Video Coding", *IEEE Transaction on Circuits and System for Video Technology*, Vol.18, No.6, pp.756-768, June 2008.

- Zhenyu Wei, Kai Lam Tang, King N. Ngan, " Implementation of H.264 on Mobile Device ", *IEEE Transaction on Consumer Electronics*, Vol.53, No.3, pp.1109-1116, August 2007.

- Zhenyu Wei, King N. Ngan, " Spatio-temporal Just Noticeable Distortion Profile for Grey Scale Image/Video in DCT Domain", *IEEE Transaction on Circuits and System for Video Technology*, Accepted for Publication.

- Zhenyu Wei, Feng Wu, King N. Ngan, " Image Compression by Inverse and Forward Texture Synthesis", *IEEE Transaction on Circuits and System for Video Technology*, Under Review.

- Zhenyu Wei, King N, Ngan, "The Perceptually Transparent Coding for Image", *IEEE Transaction on Circuits and System for Video Technology*, Under Review.

Conference Papers

- Zhenyu Wei, King N. Ngan, "A Temporal Just-noticeable Distortion Profile for Video in DCT Domain ", *IEEE International Conference on Image Processing'08*

(ICIP 2008), San Diego, USA, Oct 12-15, 2008.

- Zhenyu Wei, King N. Ngan, " Spatial Just Noticeable Distortion Profile for Image in DCT Domain ", *IEEE International Conference on Multimedia and Expo'08 (ICME 2008)*, Hannover, Germany, Jun 23-26, 2008.

- Zhenyu Wei, King N. Ngan, Hongliang Li, " An Efficient Intra Mode Selection Algorithm For H.264 Based On Fast Edge Classification " ,*IEEE International Symposium on Circuits and System'07 (ISCAS 2007)*, New Orleans, USA, May 27-30, 2007.

- Zhenyu Wei, King N. Ngan, "A Fast Rate-Distortion Optimization Algorithm For H.264/AVC ", *IEEE International Conference on Acoustics, Speech, and Signal Processing'07 (ICASSP 2007)*, Honolulu, USA, April 15-20, 2007.

- Zhenyu Wei, King N. Ngan, "A Fast Macroblock Mode Decision Algorithm for H.264 ", *IEEE Asia-Pacific Conference on Circuits and System'06 (APCCAS 2006)*, Singapore, Dec 4-7, 2006.

- Zhenyu Wei, King N. Ngan, "Implementation of H.264 on Mobile Device ", *BJ-HK Doctoral Forum 2006*, Beijing, China, July, 2006.

- Zhenyu Wei, King N. Ngan, "The Perceptually Transparent Coding for Image", *IEEE International Symposium on Circuits and System'09 (ISCAS 2009)*, Accepted for Publication.

- Zhenyu Wei, Feng Wu, King N. Ngan, " Image Compression by Inverse and Forward Texture Synthesis", *IEEE International Conference on Image Processing'09 (ICIP 2009)*, Submitted for Publication.

Nomenclature

Abbreviations

2-D	Two-dimensional
3-D	Three-dimensional
ASP	Advanced Simple Profile
AVC	Advanced Video Coding
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable Length Coding
CBP	Coded Block Pattern
CCITT	Consultative Committee of International Telegraph and Telephone
CRT	Cathode Ray Tube
CSF	Contrast Sensitivity Function
DCT	Discrete Cosine Transform
DMIF	Delivery Multimedia Integration Framework
DMOS	Difference Mean Opinion Scores
DPCM	Differential Pulse Code Modulation
DSCQS	Double Stimulus Continuous Quality Scale
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimal Truncation
EM	Expectation Maximization
FLC	Fixed Length Coding
HDTV	High-Definition TeleVision
HF	High Frequency
HVS	Human Visual System
ICT	Integer Cosine Transform
IDP	Image-Dependent Perceptual
IEC	International Electro-technical Commission
IIP	Image-Independent Perceptual
ISDN	Integrated Services Digital Network
ISO	International Standardization Organization
ITU-T	International Telecommunication Union — Telecommunications Sector
ITU-R	International Telecommunication Union — Radiocommunication Sector
J2K	JPEG 2000
J2KL	JPEG 2000 Lossless
JBIG	Joint Bi-level Image Experts Group
JM	Joint Model

JND	Just Noticeable Distortion
JPEG	Joint Photographic Experts Group
LCD	Liquid Crystal Display
LF	Low Frequency
MAC	Multiply Accumulate
MB	MacroBlock
MC	Motion Compensation
ME	Motion Estimation
MF	Medium Frequency
MME	Minimum Matching Error
MMX	MultiMedia eXtensions
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MRF	Markov Random Field
MSE	Mean Squared Error
MV	Motion Vector
NAMM	Nonlinear Additivity Model for Masking
NHT	Non-normalized Haar Transform
PAR	Pixel Aspect Ratio
PDA	Personal Digital Assistant
PNG	Portable Network Graphics
PSNR	Peak Signal to Noise Ratio
PSTN	Public Switched Telephone Network
QP	Quantization Parameter
RDO	Rate Distortion Optimization
ROI	Region of Interest
SAD	Sum of Absolute Difference
SDTV	Standard-Definition TeleVision
SIMD	Single Instruction Multiple Data
SNR	Signal to Noise Ratio
SP	Simple Profile
SPEM	Smooth Pursuit Eye Movement
SSE	Streaming SIMD Extensions
SVC	Scalable Video Coding
UDTV	Ultra-Definition TeleVision
VCEG	Video Coding Experts Group
VLC	Variable Length Coding
VO	Video Object

Notation

λ_{mode}	Lagrangian multiplier
$\lfloor x \rfloor$	The floor function, which returns the highest integer less than or equal to x
$x\%y$	Modulo operation, which is x modulo y
$sign(x)$	The sign function, which returns the sign of x
$abs(x)$	The absolute value of x
$round(x)$	Rounding operator
$Max(\cdot)$	Maximum value operator, which returns the maximum value
$Min(\cdot)$	Minimum value operator, which returns the minimum value
R	Bit rate, specified in bits/pixel (bpp) for an images, or bits/sec (bps) for a video sequence
D	Distortion, generally measured by the mean-squared error (MSE) or sum of absolute difference (SAD)
$\|\| \cdot \|\|_2$	Order 2 norm

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Objectives

Visual information is the most important information that is perceived, recognized, and understood by human beings in the surrounding world. However, the earlier computer and communication systems mainly focused on processing and transmitting the text or speech information due to their limited process capability. With the revolution in computer and communication technologies in the recent three decades, image and video processing become possible and necessary. Today we are facing a digital world—digital networks, digital representation of images, movies, video, TV, voice, digital library—all because the digital representation of the signal is more robust than the analog counterpart for processing, manipulation, storage, recovery, copy, transmission, even across different platforms and applications. Despite many advantages of the digital representations of pictures, they need a very large number of bits for storage and transmission in raw (uncompressed) data form, as compared with other digital applications such as text or speech. For example, if we want to transmit the standard-definition television (SDTV) color video (with resolution 720×576) via a network in real time at a frame rate of 25 frames/second, we need about 248.8 million ($= 720 \times 576 \times 24 \times 25$) bits per second (Mbps) of bandwidth. This is impossible given limited bandwidth in the real world. In Table 1.1, we provide the raw data bit rates for some typical digital image and video sources.

With the growth of more complex video services such as 3-D movies, 3-D games, and ultra-definition television (UDTV), how to bridge the gap between the required huge amount of image and video data and the limited hardware capability becomes an important and urgent problem. Image and video data compression techniques give us a promising way to solve above the problems. Data compression is a technique to reduce

Source		Frame Rate	Resoultion	Bit Rates of Raw Data
Image	Standard-resolution	N/A	512×512	6.3 Mbits
	High-resolution	N/A	2048×2560	125.8 Mbits
Video	Video Phone	7.5	128×96	2.2 Mbps
	Video Conferencing	15	352×288	36.5 Mbps
	VCR	25	352×288	53.8 Mbps
	SDTV (PAL)	25	720×576	248.8 Mbps
	SDTV (NTSC)	30	720×480	248.8 Mbps
	HDTV (720P)	60	1280×720	1.3 Gps
	HDTV (720I)	30	1280×720	663.5 Mbps
	HDTV (1080P)	30	1920×1088	1.5 Gps
	HDTV (1080I)	30	1920×1088	1.5 Gps

Table 1.1*: Raw data bit rates for some typical digital image and video sources.*

the redundancies in data presentation in order to decrease data storage requirements and communication costs. It is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. Therefore, the development of image and video data compression techniques plays a vital role for future communication systems and advanced multimedia applications.

Interestingly, image and video contain significant amount of superfluous and redundant information, including the spatial redundancy that stands for the correlation among neighboring pixels in one frame, and the temporal redundancy that is related to the correlation among the consecutive frames in one video sequence. By eliminating these redundancies, image and video data compression can be achieved. In the past two decades, image and video compression techniques have made a rapid progress, and several international image and video compression standards have been finalized. The widely used standards include JPEG/JPEG 2000 [ITU-T and ISO/IEC, 1993b; ISO/IEC, 2004a] for lossy/lossless still image compression, H.261/H.263 [ITU-T, 1994; ITU-T, 2001] for video phone and conferencing, MPEG-1 [ISO/IEC, 1992] for CD-ROMs storage, MPEG-2 [ITU-T, 1995] for digital TV broadcasting, and MPEG-4 [ISO/IEC, 1999] for multimedia distribution. Recently, ITU-T and ISO jointly defined a new video coding standard named H.264/AVC [ITU-T and ISO/IEC, 2005], roughly doubling the coding performance compared with H.263.

Although picture compression offers numerous advantages and it is the most sought-after technology in many multimedia application areas, it still has some deficiencies. Currently, it is more and more difficult to improve the coding performance. In order

to increase the compression efficiency, many complicated techniques have been introduced in compression standards. Consequently, small improvements are accomplished at the expense of increasing complexity of encoder and decoder. Some studies show that JPEG 2000 is more than 30 times more complicated compared to baseline JPEG, and the complexity of H.264/AVC is about 5-8 times compared to H.263. Extra complexity incurred by new compression techniques is one of the most serious drawbacks of picture compression, which discourages its usage in some areas (e.g. in many real time multimedia communication applications). In many hardware and systems implementations, the extra complexity can increase the system cost and reduce the system efficiency, especially in the areas of applications that require very low-power consumption. Thus, it is very necessary to propose more efficient compression techniques to reduce the complexity of current image and video compression technologies, especially for state-of-the-art standards such as H.264/AVC.

In addition, the current mainstream compression schemes are signal processing based. Most of these techniques treat images and videos as 2-D or 3-D signals. In these methods, only statistical properties among pixels are considered, and the perceptual features are often neglected. So they could not explore the perceptual properties existing in the images very well. For example, the pixel-wise distortion metric, such as mean squared error (MSE) and peak signal-to-noise ratio (PSNR) are widely adopted in the existing coding standards. But in the meantime, they have been criticized for not correlating well with perceived quality measurement. Since the human eyes are the ultimate receiver of the majority of processed images and videos, it is very important to exploit the perceptual redundancy in image and video compression algorithms. The removal of the perceptual redundancy has many advantages. First, it ensures that only the visually important information is encoded or protected. Secondly, better compression performance can be achieved by discarding perceptually unnecessary information. Currently, the rapid growth of the research on human visual system (HVS) and computer vision techniques gives us two promising directions to develop perceptual picture compression technology.

Facing with the problems discussed above, the objective of this thesis is to develop more efficient and perceptual image and video coding techniques. On the one hand, the

complexity of the picture compression can be reduced significantly with little degradation of the coding performance; on the other hand, the perceptual redundancy existing in pictures is also investigated thoroughly.

In the following part of this chapter, some background knowledge related to this thesis is introduced. In Section 1.2, the development of image and video coding is briefly reviewed. It could be observed that although the performance of the picture coding technology continues to be improved, the techniques adopted in those international picture coding standards become more and more complex. Section 1.3 presents the features of human visual system, which is the basis of the perceptual picture coding techniques studied in this thesis. The outline of this thesis is given in Section 1.4.

1.2 Development of Image and Video Coding

Picture compression has been developed for many years and become an integrated part of today's digital communications system — digital telephony, facsimile, digital mobile communication, video conferencing, Internet, broadcasting, etc. Other applications include image archival system, digital library, DVD, movie and video distribution, film industry, to name a few. Picture coding techniques impact our life deeply. The number of applications will continue to grow. As a result, it is very necessary to establish standards for common picture compression systems to be perfectly interoperable in different systems and platforms.

Now there are mainly two international standardization organizations to define standards for picture compression. One is International Telecommunication Union — Telecommunications Sector (ITU-T) formerly known as Consultative Committee of International Telegraph and Telephone (CCITT), which principally deals with information transmission. Another is International Standardization Organization (ISO), which deals with information-processing related issues, such as picture storage and retrieval. In the past several decades, these two groups have released plenty of standards for still image and video coding. With the growth of more and more advanced techniques, the performance of picture coding standards has been improved significantly, but the complexity of them has remarkably increased as well.

1.2.1 Still Image Coding Standards

In this section, some still image coding standards are introduced, including lossy and lossless image coding.

JPEG

JPEG [ITU-T and ISO/IEC, 1993b] is the standard jointly developed by ISO and ITU-T in 1992 for the compression of continuous-tone (gray-scale or color) still images, which is officially named as ISO/IEC IS (International Standard) 10918-1: *Digital Compression and Coding of Continuous-tone Still Images* or also ITU-T Recommendation T.81. The JPEG standard describes a family of image compression techniques. The compression ratio can be adjusted by considering the tradeoff between bit-rate and image quality. JPEG typically can achieve 10:1 compression with little perceptible loss in image quality.

JPEG provides four modes of operation, sequential (baseline), hierarchical, progressive, and lossless. In general, the JPEG has the following features: resolution independence, no absolute bit-rate targets, luminance-chrominance separability, and extensibility. In the four modes defined in JPEG, the baseline JPEG has been widely used for image compression, which can compress still image with bit-rates of 0.25-2 bits per pixel. It can be summarized in three steps: (1) DCT computation; (2) Quantization; (3) Entropy coding. JPEG standard contains a lossless mode, but it uses a completely different technique from the lossy JPEG standard and also is not compatible with the bitstream syntax of the lossy part.

JPEG has very low complexity and is very easy to implement. However, the blocky artifacts are introduced when the compression ratio is high, which is very annoying and reduces image quality significantly.

JPEG can be used to compress video sequences by considering video as a sequence of still image frames. This method is so-called Motion JPEG. Although it is not defined in the standard, it has been popularly used in industry due to its low computational load.

JPEG 2000

JPEG 2000 [ISO/IEC, 2004a] is a wavelet-based image compression standard created by JPEG committee in the year 2000. Although baseline JPEG has been very popular in marketplace for more than a decade, it has many drawbacks. JPEG can supply good picture quality with high bit-rates, but the perceptual quality of compressed image declines significantly with the reduction of bit-rate (less than 0.25 bpp). Thus JEPG is not suitable for the bandwidth-constrained networks. To improve JPEG standard, JPEG committee began to design a totally novel still image compression standard since 1997, which is called JPEG 2000. In 2000, the Part 1 of JPEG 2000 standard was finalized as an International Standard ISO/IEC 15444-1:2000, which describes the core coding system and is the most important part of JPEG 2000.

JPEG 2000 incorporates Discrete Wavelet Transform (DWT), Embedded Block Coding with Optimal Truncation (EBCOT), and other latest image compression techniques to provide a unified optimized framework to achieve both lossless and lossy compression using the same algorithm and the bitstream syntax. JPEG 2000 not only improves the compression performance compared with baseline JPEG but also is optimized for scalability and interoperability in networks and noisy mobile environments. It can achieve up to about 20% compression ratio gain for medium compression rates in comparison to the baseline JPEG standard. For the lower or higher compression rates, the improvement can be somewhat greater.

The main drawback of the JPEG 2000 standard compared to current JPEG is that the coding algorithm is too complex and the computational loads are much higher. Literatures [Skodra et al., 2000; Santa-Cruz and Ebrahimi, 2000] show that JPEG 2000 is more than 30 times complex as compared to baseline JPEG. Moreover, for the low bit-rate compression, JPEG 2000 also introduces blurring artifacts.

JBIG and JBIG2

JBIG [ITU-T and ISO/IEC, 1993a] is a lossless image compression standard for black-and-white image designed by the Joint Bi-level Image Experts Group in 1994, standardized as ISO/IEC standard 11544 and as ITU-T recommendation T.82. JBIG is also known as JBIG1, which is widely utilized for compression of binary images, particularly for faxes, and other images.

The Joint Bi-level Image Experts Group defined a new bi-level image compression standard called JBIG2 [ITU-T and ISO/IEC, 2001], which is an improvement of JBIG and is suitable for both lossless and lossy compression. In its lossless mode, JBIG2 typically generates files with one half to one quarter the size of JBIG. JBIG2 has been standardized in 2000 as the international standard ITU T.88, and in 2001 as ISO/IEC 14492.

JPEG-LS

JPEG-LS [ITU-T and ISO/IEC, 2003] is the new lossless/near-lossless compression standard for continuous-tone images, standardized as ISO/IEC standard 14495-1 and as ITU-T recommendation T.87. The standard is based on the LOCO-I algorithm (LOw COmplexity LOssless COmpression for Images) [Weinberger et al., 1996] developed by Hewlett-Packard Laboratories.

JPEG-LS offers a lossy mode of operation, termed near-lossless, where the difference between every sample value in a reconstructed image component and the corresponding value in the original image is controlled by a maximum error threshold d. In fact, the lossless mode is just a special case of near-lossless compression, with $d = 0$. So far, JPEG-LS is the best lossless image compression standard. It has very low complexity and can offer better compression efficiency than lossless JPEG, and even better than lossless JPEG 2000.

1.2.2 Video Coding Standards

From the 1980s of last century, ISO/IEC and ITU-T continued to define a series of digital video coding international standards for different application fields, including the MPEG family by ISO/IEC and the H.26x family by ITU-T. Figure 1.1 illustrates the development of the international video coding standards.

H.261

H.261 [ITU-T, 1994], with a full name "Video Codec for Audiovisual Services at p×64kbit/s", is designed by ITU-T in 1990 for video conferencing over ISDN (Integrated Services Digital Network) lines. H.261 was the first truly practical digital video coding standard (in terms of product support in significant quantities). In fact, all

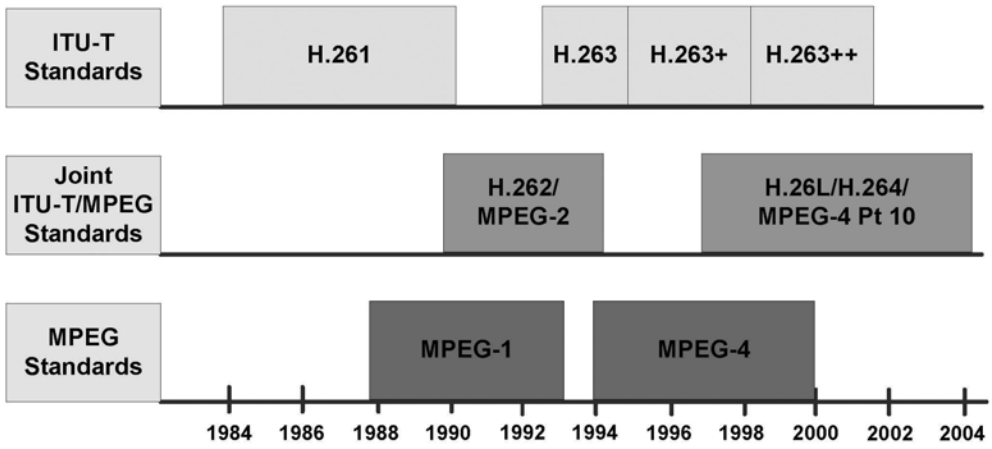

Figure 1.1*: International video coding standards timeline.*

subsequent international video coding standards are based closely on the framework of H.261. H.261 is also called "p×64kbit/s" standard. The reason is that H.261 is designed for the channel where data rates are multiples of 64 kbit/s. p takes integer value ranging from 1 to 30.

H.261 adopts many advanced video coding techniques, including block-matching motion estimation, integer-pel resolution motion compensation, only one forward reference frame. It also uses an 8×8 DCT for each frame to reduce spatial redundancy, and a DPCM (differential pulse code modulation) loop to reduce temporal redundancy.

The H.261 standard actually only specifies how to decode the video. Encoder designers are left free to design their own encoding algorithms, such as motion estimation, rate control, etc., as long as their output bitstream can be decoded by any decoder made according to the standard. This not only ensures that the bitstreams produced by different manufacturers and coding systems are fully compatible, but also gives free space for manufacturers to develop their own key encoding techniques. It is also adopted by all subsequent video coding standards. Therefore, H.261 remains a major historical milestone in the development of the video coding standards.

MPEG-1

MPEG-1 [ISO/IEC, 1992], with a full name "Coding of Moving Pictures and Associated Audio - for Digital Storage Media at up to about 1.5Mbit/s", was designed for progressively scanned video used in multimedia applications, and the target was to compress VHS-quality raw digital video and CD audio down to 1.5 Mbit/s without

excessive quality loss, targeting applications in Video CDs, digital cable/satellite TV, etc.

Compared with H.261, MPEG-1 uses two important techniques: bi-directional motion estimation and half-pel resolution motion compensation.

Bi-directional motion estimation allows the use the forward and backward frame as the reference frames. Thus, MPEG-1 contains three types of frames: I- (intra frame), P- (inter frame) and B-frames (bi-directional frame). B-frames can further reduce the bit-rate of output stream. However, the complexity of bi-directional motion estimation is higher, and the coding order is also different from the scan order. Therefore, frame re-ordering is needed at the encoder and the decoder.

Half-pel motion compensation is another key technique in MPEG-1. This method is to interpolate the luminance and chrominance value of each sub-pixel according to the related integer pixel values. The performance of the encoder is improved significantly, however, extra computational load is also introduced.

It should be pointed out that video coding part is only one part of MPEG-1 standard. Besides the video part, it also includes the systems, audio, conformance testing and reference software.

MPEG-2

MPEG-2 [ITU-T, 1995], with a full name "The generic coding of moving pictures and associated audio information", is targeted at TV studios and TV broadcasting for standard TV and HDTV (high definition TV). MPEG-2 is jointly developed by ISO and ITU-T. The video part of MPEG-2 is formally known as ISO/IEC 13818-2 and as ITU-T Recommendation H.262.

Compared with MPEG-1, the major improvement of MPEG-2 includes two techniques: support for interlaced video coding and the scalable video coding.

In order to reduce the amount of data, TV normally adopts a format called "interlaced video", which separates the picture into two fields: the "top field" and the "bottom field". The two fields are displayed alternately. For the non-interlaced video, it is called "progressive video". MPEG-2 supports both formats.

Scalable video coding (SVC) is intended to accommodate decoding the video for various applications with diverse needs in quality, frame rate and resolution all from a

single compressed bitstream. MPEG-4 supports various modes of scalability, including spatial scalability, temporal scalability and SNR (signal to noise ratio)scalability.

MPEG-2 defines a complete system from encoding to transmission, and the supported bit-rates cover a very wide range. The application field of MPEG-2 is broad enough to cover the satellite broadcasting service, cable television, digital terrestrial television, e-cinema, home theater, interactive media, remote video surveillance and so on. DVD, a popular optical disc storage media format, is also based on the MPEG-2 standard. We can say that MPEG-2 is a very successful video coding standard.

H.263

H.263 [ITU-T, 2001] is a video coding standard designed as a low-bitrate compressed format for videoconferencing over PSTN, ISDN, Internet and other networks. It was developed by the ITU-T Video Coding Experts Group (VCEG) and finalized as one member of the H.26x family of video coding standards in 1996.

As the public switched telephone network (PSTN) and wireless network have very limited bandwidth and high error rate, ITU-T further designed the enhanced versions of H.263 known as H.263v2 (also known as H.263+ or H.263 1998) and H.263v3 (also known as H.263++ or H.263 2000) to satisfy the requirement of high compression efficiency and strong ability of error resilience.

Many concepts originally proposed by H.263, such as variable block size motion estimation, motion vector prediction, unrestricted motion estimation, multi-reference frame motion compensation and so on, were subsequently adopted by other later standards.

MPEG-4

MPEG-4 [ISO/IEC, 1999] is a video compression technology developed by MPEG. Its target applications include Internet, multimedia, interactive video games, personal communications, multimedia messaging, network database services, remote video surveillance, wireless multimedia, and so on.

Currently, MPEG-4 contains a number of parts, including system, visual, audio,

conformance testing, reference software model, delivery multimedia integration framework (DMIF), optimized reference software, carriage on IP networks, hardware reference, advanced video coding (AVC), and so on. AVC is also known as the H.264 video coding standard, jointly developed by ITU-T and MPEG.

The most important contribution of MPEG-4 is to introduce the concept of object-based video coding. A scene consists of several video objects (VOs). The composition of VO depends on the specific application and the actual environment. VO can be a rectangular frame same as the traditional standards, or a object with arbitrary shape segmented from the frame. Each VO uses three types of information to describe: motion, shape and texture.

In addition, MPEG-4 also introduces a number of advanced techniques, such as wavelet transform for video, Sprite coding, zero-tree scanning, and so on. However, many MPEG-4 tools, such as object-based video coding techniques, are not widely applied in practice due to the restriction of complexity. The popularly used MPEG-4 profiles are the Simple Profile (SP) and the Advanced Simple Profile (ASP). The former one is very similar to H.263 and the latter is to introduce 1/4-pel resolution motion compensation and global motion estimation techniques to H.263.

H.264/AVC

H.264/AVC [ITU-T and ISO/IEC, 2005] is the latest video coding standard jointly developed by the ITU-T VCEG and the ISO/IEC MPEG. The main goals of the H.264/AVC standardization are to enhance compression performance and provide a "network-friendly" video representation for "conversational" (video telephony) and "non-conversational" (storage, broadcast, or streaming) applications. H.264/AVC presents a number of advances in standard video coding technology, in terms of both coding efficiency enhancement and flexibility to effectively use over a broad variety of network types and application domains. The features of the new design provide approximately a 50% bit rate saving for equivalent perceptual quality relative to the performance of prior standards. H.264 has some features listed as below:

- Low bit-rate, high quality
- Wide application fields

- Robust (error resilient) video transmission
- Network friendliness

In order to achieve the above features, H.264 uses many advanced techniques. One of them is the intra prediction. It uses the spatial relationship to reduce redundancy. There are 13 intra modes: 4 for 16×16 block size, and 9 for 4×4 block size in the baseline profile.

H.264 also uses many motion compensation techniques. One is the variable block sizes. It has 7 different block sizes, from 4×4 to 16×16. When coding, it searches these modes one by one and chooses the best one as the final mode for current macroblock. For example, for the high-detailed block, small size mode will be used. For some background with no motion, the encoder will choose 16×16 block size to code it. In the previous video coding standards, only one or two block sizes are used. It also adopts 1/4 resolution sub-pixel motion compensation. In order to eliminate blocking artifacts, an effective deblocking filter is used. It also utilizes multiple reference frames and the number of reference frames can be from 1 to 16. Previous video standards only use one reference frame, so H.264 can utilize temporal correlation efficiently to remove temporal redundancy.

H.264 uses the integer ICT (Integer Cosine Transform) to speed up the transform speed. ICT uses the left or right shift to replace multiplication and division and it is very easy to implement. H.264 also adopts new entropy coding methods, CABAC (context-based adaptive binary arithmetic coding) and CAVLC (context-adaptive variable length coding).

By using these techniques, H.264 achieves a better performance than the previous video coding standards. The encoding quality is improved, but the complexity of the encoder and computational cost are increased as the cost. Some literatures show that complexity of H.264 is about 5-8 times that of H.263, so it is very difficult to implement real-time H.264 codec.

1.3 Overview of Human Visual System

Vision and hearing are the two most important means by which humans perceive the outside world. 80% $\sim$ 90% of all neurons in the human brain are estimated to be

involved in visual perception [Young, 1991]. Since the human visual system (HVS) is the final receiver of most visual signals, the features of the HVS have been studied for many years. It can be subdivided into two major components: the eyes, which capture the light and convert it into signals that can be understood by the brain, and the visual pathways in the brain, via which the signal can be transmitted and processed [Winkler, 2000]. Although the HVS is a very complicated system, some important properties have been investigated via extensive vision experiments. These visual properties can explain some important phenomena of human perception and are also relevant to image and video processing.

This section will introduce some important perceptual characteristics that are relevant to the models and algorithms discussed in this thesis. These vision features have played essential role in perceptual image/video coding and picture quality metric. They are: *luminance adaptation*, *spatio-temporal contrast sensitivity function*, *contrast masking* and *gamma correction*.

1.3.1 Weber-Fechner's Law and Luminance Adaptation

Human eyes are capable of perceiving an enormous range of light intensities. The human capacity to distinguish a change in the magnitude of a stimulus depends on not only the quantity of the change, but also the magnitude of that stimulus. For example, it is easy to distinguish a 1-kg weight from a 2-kg weight, but it is difficult to distinguish a 100-kg weight from a 101-kg weight. This property is known as the *Weber-Fechner law* [Nettravali and Haskell, 1988], which claims that noticeable intensity difference is proportional to the background intensity value over a wide range of intensities, as shown in Figure 1.2 [Cornsweet, 1970]. It can be expressed as

$$C = \frac{\Delta L}{L} \tag{1.1}$$

where ΔL and L stand for the intensity change and the background intensity in cd/m^2, respectively. C is the threshold contrast and remains nearly constant over a wide range of intensities (from faint lighting to daylight).

Observed from Figure 1.2, Weber-Fechner's Law is valid for a wide range of background luminance. Outside of this range, the intensity discriminative ability deteriorates. Since 8-bit digital image scales from 0 to 255, it is possible that image contains

very dark or very bright areas where the Weber-Fechner's Law is invalid. For the very dark or very bright regions, the visibility threshold is higher. Based on the above observations, the visibility threshold should be adaptive to the background luminance. This effect is called luminance adaptation.

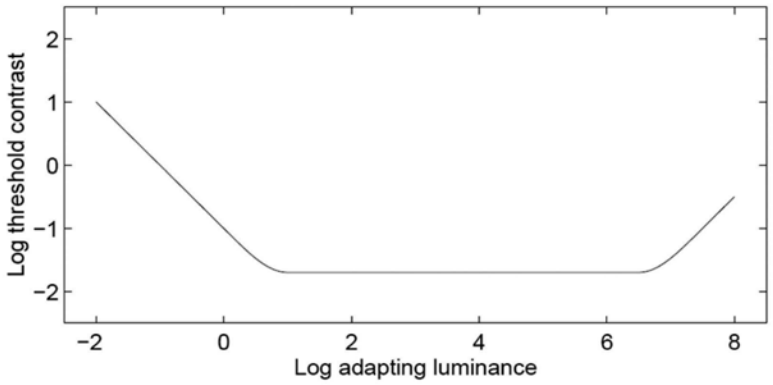

Figure 1.2: *Illustration of the Weber-Fechner law [Cornsweet, 1970].*

1.3.2 Spatio-temporal Contrast Sensitivity Function

A very convenient way of working in a linear-system kind of form is to characterize the response of the considered system with respect to a set of harmonic functions. This is also done for the description of the visual system and leads to the concept of contrast sensitivity function (CSF). The spatio-temporal contrast sensitivity function describes the relationship of human eye's sensitivity versus spatial and temporal frequency. It is well known that the eye is more sensitive to the lower spatial frequencies than to the higher ones. The sensitivity of the visual system changing over time is also related to the perception of object motion. The CSF represents the neurons' contrast threshold as a function of the harmonic function, which is actually a multivariate function of the spatial frequency, the temporal frequency, the orientation and the color component.

Robson [Robson, 1966] first explored the relationship between the spatial and temporal contrast sensitivity using temporally modulated counterphase gratings at a variety of the spatial and temporal frequencies. The results are shown in Figure 1.3.

In Figure 1.3, the data are presented in two ways. Figure 1.3A shows the spatial CSFs for different temporal frequencies ranging from 1 to 22 Hz. It could be observed that the spatial CSF changes from the band-pass to the low-pass with the increase

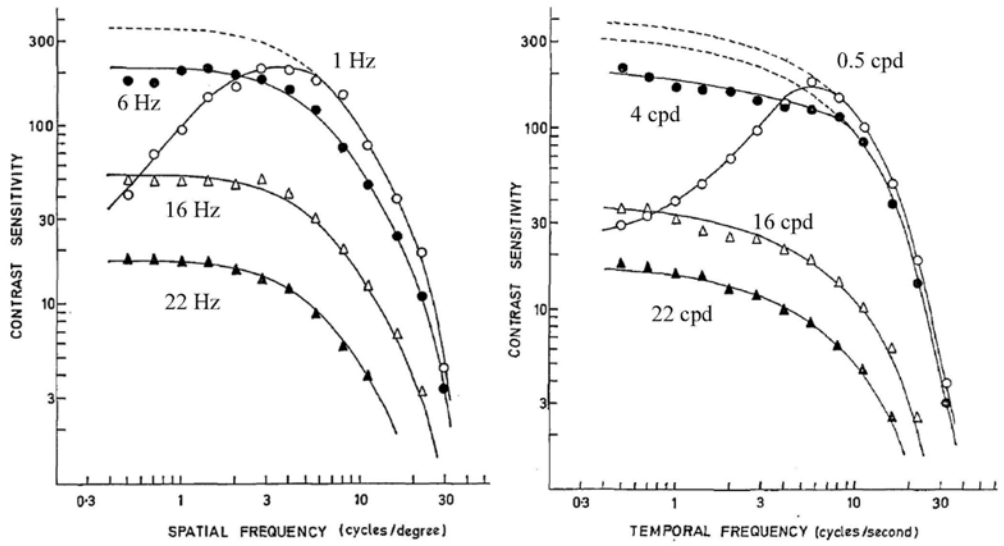

Figure 1.3: *(A) Spatial contrast sensitivity function for different temporal frequencies. (B) Temporal contrast sensitivity function for different spatial frequencies [Robson, 1966].*

of the temporal frequency. This indicates that the CSF for the low spatial frequency depends on temporal frequency. Figure 1.3B shows the temporal CSFs for different spatial frequencies ranging from 0.5 to 22 cpd (cycle per degree). Here we can see that the temporal CSF is band-pass at the low spatial frequencies and low-pass at the higher spatial frequencies. This also indicates that CSF for the low temporal frequency is spatial frequency dependent. It is remarkable to note the similarities between the spatial and the temporal mechanism shown in Figure 1.3. The spatio-temporal CSF shows a non-separable property for the spatial and temporal frequency. Families of the curves in Figure 1.3 can be accumulated to produce a spatio-temporal contrast sensitivity contour. In Figure 1.4, the CSF is plotted as a function of both temporal and spatial frequency.

Literatures have shown that the spatio-temporal CSF curve not only varies with the background luminance, but also exhibits different properties for different color channels [Peterson et al., 1993b]. Moreover, contrast sensitivity is also related to the eye movement and the velocity of the observed moving objects [Daly, 1998]. These findings are very important to improve the human visual system model and will be discussed in details in the following chapters.

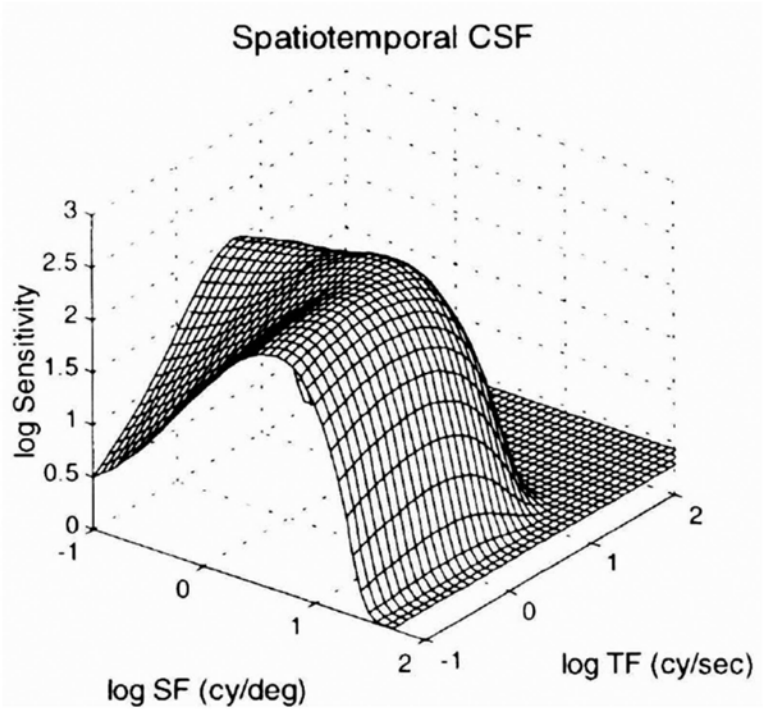

Figure 1.4: *A presentation of a spatio-temporal contrast sensitivity surface [Daly, 1998].*

1.3.3 Contrast Masking

Contrast masking is a very important phenomena in vision and in image processing as it describes the interactions between stimuli, which refers to the reduction in the visibility of one stimulus (the target) in the presence of another one (the marker). A well-cited model of contrast masking was proposed by Legge and Foley [Legge and Foley, 1980; Legge, 1981].

Figure 1.5 shows an example of typical masking curves. The horizontal axis represents the log of the masker contrast C_M, and the vertical axis is the log of the target contrast C_T at the detection threshold. C_{T_0} stands for the detection threshold for the target contrast without any masker. If the masker contrast is larger than C_{M_0}, the target contrast value will grow with increase of the masker contrast.

Two cases can be observed in Figure 1.5. For case A, the masker and the target stimuli have different characteristics, so masking is the dominant effect. For case B, the masker and the target stimulus have similar characteristics. The detection threshold of the target decreases (facilitation) first and then increases with the increase of the masker contrast. The facilitation is a very special phenomenon in contrast masking

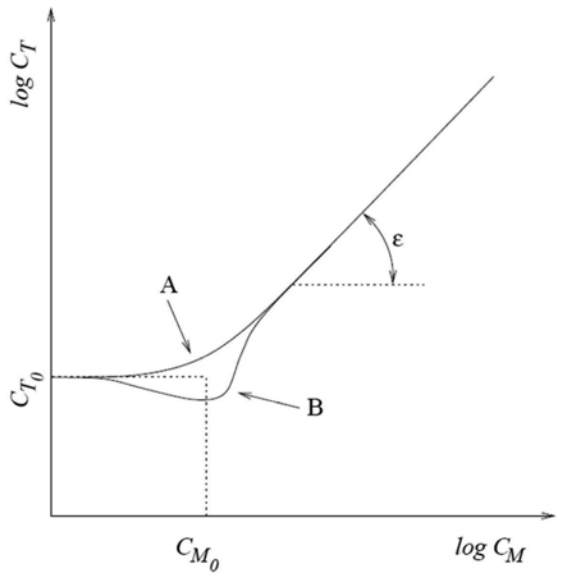

Figure 1.5: *Illustration of typical masking curves. Curve A results for stimuli with different characteristics. B for stimuli with similar characteristics*

discrimination, which implies that the target is easier to perceive due to the presence of the masker in a certain contrast range.

In some HVS models [Watson, 1993; Hontsch and Karam, 2002; Zhang et al., 2005; Jia et al., 2006], it is assumed that there is no masking effect when the contrast is lower than a certain value. For the increasing part of the contrast masking curve, Legge's power law [Legge, 1981] is used to represent the relationship between the increment threshold ΔC and the background contrast C:

$$\Delta C = kC^{\epsilon} \tag{1.2}$$

The exponent ϵ is the slope in Figure 1.5, and k is a sensitivity parameter.

1.3.4 Gamma Correction

Digital images and videos are stored, transmitted and processed based on the intensity of each pixel. Usually, they will be displayed on the CRT (Cathode Ray Tube) and LCD (Liquid Crystal Display) monitors. However, for displays, the brightness (in cd/m^2) and the pixel values ($0 \sim 255$ for 8-bit images) have a non-linear relationship (as shown in Figure 1.6). To correct the non-linearity, *gamma correction* formulated as Eq. (1.3) has to be carried out,

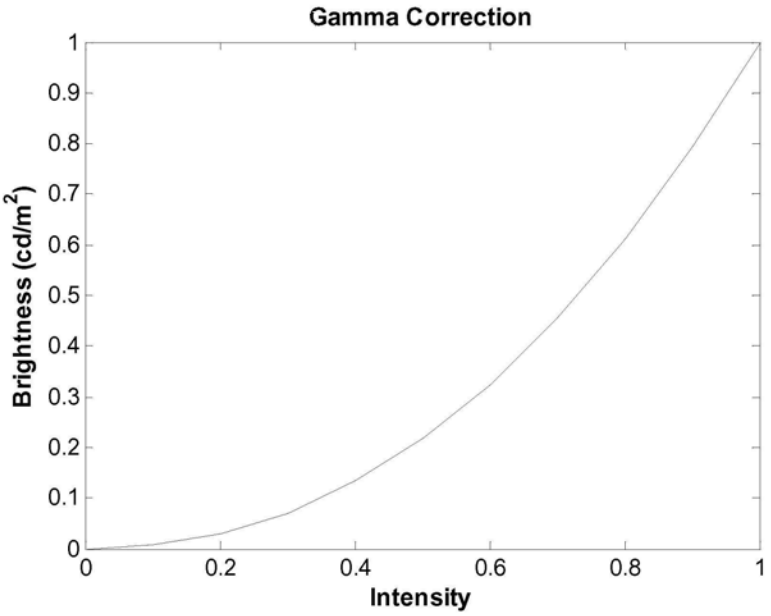

Figure 1.6: *Normalized gamma correction curve ($\gamma = 2.2$)*

$$L = cI^{\gamma} \tag{1.3}$$

where L is the brightness value in cd/m^2, I is the pixel intensity value and ranges from 0 to 255 for 8-bit images. γ is the correction parameter and may vary for different monitor settings. Usually, $\gamma = 2.2$ for CRT display.

1.4 Thesis Outline

This thesis focuses on the efficient and perceptual coding techniques for image and video, and is organized into two parts accordingly:

- Part I. Efficient Picture Coding Techniques
- Part II. Perceptual Picture Coding Techniques

Part I discusses the subject of the efficient coding techniques for H.264. In Chapter 2, an efficient intra mode selection algorithm for H.264 is presented, which is based on edge classification and rate-distortion estimation. We first introduce the background of this research topic and give a brief overview of intra mode decision and RDO (rate distortion optimization) in H.264. Then a fast intra model decision algorithm is proposed, where an edge detection method is applied to speed up the intra mode decision

significantly by using non-normalized Haar transform (NHT). Since RDO is another very important, but time-consuming technique adopted in H.264, we presented a fast RDO method, based on three key techniques: precise bit-rate estimation model, decomposition of luma and chroma RDO computation, and distortion computation in transform domain. The performance of the proposed algorithm is evaluated against the state-of-the-art techniques via extensive coding experiments.

H.264 allows variable block size in inter frame coding. In previous video coding standards such as H.263, MPEG-1, MPEG-2, only 1 or 2 block sizes are allowed. So it is obvious that the 7 different block size modes consume the main computational cost of the H.264 encoder. In Chapter 3, we address the solutions to this problem and propose a fast inter mode decision algorithm for H.264. First, the variable block size in H.264 is briefly introduced. Then, our proposed fast inter mode decision algorithm is presented, which is based on following techniques: Pskip mode early detection based on transform domain, mode prediction method, early termination technique, post-search technique and intra mode skip detection. The performance of the proposed algorithm is evaluated against two famous fast inter mode decision techniques — Ahmad's [Ahmad et al., 2004] and Yu's [Yu et al., 2006] method.

Since a real-time H.264 codec on embedded system has very broad applications, Chapter 4 discusses the implementation and optimization of H.264 baseline profile on mobile device. The performance and technical features of the new generation international video coding standard H.264 and the powerful embedded processor PXA27x produced by Intel are introduced first. Then implementation of H.264 on mobile device is presented, which includes profile selection, code porting and code optimization. For the optimization part, three levels of the optimization work are explained: the program level, the algorithm level and the instruction level. The performance of the optimized encoder and decoder are experimentally demonstrated on the mobile device. Finally, an H.264 based mobile video conferencing system is implemented.

Part II investigates the subject of various techniques in perceptual picture coding. Since HVS (human visual system) plays a vital role in perceptual picture coding and JND (just noticeable distortion) is a good solution to model the HVS, a DCT (discrete cosine transform) based spatio-temporal JND model for grey scale picture is presented in Chapter 5. The definition and the classification of JND model are briefly introduced

first and several well-known JND models are reviewed in this chapter. Our proposed JND model incorporates the spatial CSF (contrast sensitivity function), the temporal CSF, the luminance adaptation effect, and the contrast masking effect based on block classification. The proposed JND model is compared with several well-cited JND models to evaluate its performance. Extensive experiments on different resolution images and videos have demonstrated the advantages of our proposed JND model.

In Chapter 6, the proposed JND model is applied in perceptual picture coding. We first implement a perceptually transparent image compression method for monochromatic images, where the quantization factor can be tuned for each block according to the JND threshold to make the quantization error invisible. The proposed algorithm is also extended to the color images and evaluated against other state-of-the-art lossless and near-lossless codecs. Furthermore, a perceptual transparent H.264 codec for videos is presented, based on the proposed spatio-temporal JND model. Compared with the original H.264 encoder, this codec can reduce bit-rate significantly with the same visual quality.

In Chapter 7, we exploit another kind of perceptual picture coding technique — computer vision based image compression. We first introduce a new technique named inverse texture synthesis, which can convert texture to a small sample that includes almost all features of the input texture without resolution degradation. A novel image compression scheme based on inverse and forward texture synthesis is presented. The superiority of the proposed compression scheme against JPEG and JPEG 2000 is experimentally demonstrated in terms of bit-rate and perceived quality.

The thesis is concluded in Chapter 8, where the contributions of this thesis are summarized and the future research directions are discussed.

Part I

EFFICIENT PICTURE CODING TECHNIQES

Chapter 2

Fast Intra Mode Decision Algorithm for H.264

2.1 Introduction

H.264 is the latest standard for moving picture coding [ITU-T and ISO/IEC, 2005]. In order to achieve outstanding coding performance, many advanced techniques are used, such as: intra mode decision, variable block size motion estimation, 1/4 pixel resolution motion estimation, multiple reference frames, deblocking filter, integer cosine transform (ICT), CABAC and CAVLC entropy coding, etc. It is shown that these techniques can provide nearly 50% bit rate reduction compared with other previous standards. However, as the computational complexity of the encoder increases, reducing the complexity of some of its algorithms becomes an important and challenging task.

Among the new techniques introduced by H.264, intra mode plays a vital role because it can reduce spatial redundancy substantially. In contrast to some previous video standards, such as MPEG-4 [ISO/IEC, 1999], where intra prediction is performed in transform domain, in H.264 intra mode, prediction block is formed based on neighboring reconstructed pixels and is subtracted from current block before encoding. In luma component, intra prediction is applied for each 4×4 block and for a 16×16 macroblock, and also for each 8×8 block in high profile. There are nine modes for 4×4 luma block, four modes for 16×16 luma block and four modes for 8×8 chroma block. In order to attain the best coding performance, a very time-consuming technique named RDO (rate distortion optimization) is used. It computes the real bit-rate and distortion between original and reconstructed frames for each mode. Then the RDcost (rate distortion cost) based on Lagrangian rate distortion formula is calculated. The mode which has the minimum RDcost will be chosen as the final coding mode. Therefore, the computational load of this kind of exhaustive searching algorithm is not acceptable for real-time applications. So, it is very important to design a fast intra mode decision method to

reduce the complexity of encoder.

In recent years, several fast intra prediction mode selection algorithms have been proposed [Pan and Lin, 2004; Pan and Lin, 2005; Meng and Au, 2003; Yang and Po, 2004; Kim et al., 2006; Yu et al., 2006; Cheng and Chang, 2005]. From the definition of intra modes, it has been found that they have very strong directionality. Each mode produces prediction block along corresponding direction. If the block edge can be detected and the pixels are predicted along the direction of block edge, a good prediction result can be obtained. Pan and Lin [Pan and Lin, 2005] presented an edge detection method to speed up the intra mode decision by using Sobel operator. However, since Sobel operator should be operated on each pixel and the edge direction histograms also need to be calculated, the extra computation load is still high. Li and Ngan [Li and Ngan, 2006] introduce a fast and efficient edge detection method which is based on the properties of coefficients in non-normalized Haar transform (NHT). Since only a small number of addition and subtraction operations are involved and there are up to 42 models in this method, block edge can be detected very quickly and accurately. Based on the block edge information, only few modes need to be checked in intra prediction.

Because RDO needs to perform transform and entropy coding for each mode to get the real distortion and bit rate, lots of computational costs are introduced. Therefore, if a good method is designed to predict the bit rate and distortion accurately, plenty of unnecessary computation, such as entropy coding, inverse transform, etc, will be reduced. In order to solve this problem, several methods can be used. Because of conservation of energy, the distortion can be calculated in transform domain. But for bit-rate estimation, since entropy coding in H.264/AVC is more complex than that in previous standards, it is not easy to get accurate bit-rate by using lookup tables for runs and levels. Some previous literatures have already studied the bit-estimation problem. Chiang and Zhang [Chiang and Zhang, 1997] and Corbera and Lei [Ribas-Corbera and Lei, 1999] proposed rate-distortion models based on quantizer-domain. The ρ-domain rate model in [He et al., 2001; He and Mitra, 2002; He et al., 2002] investigates the linear relationship between rate and non-zero quantized DCT coefficients. But unfortunately, these methods were designed for rate control originally and more suitable for bit-estimation in frame level, but not in macroblock level. Chen and He [Chen and

He, 2004] proposed a bit-rate estimation model, but since the parameters of this model are fixed and not self-adaptive, the estimation accuracy is limited. The method in [Tu et al., 2006] mainly focuses on the inter mode decision, but in H.264/AVC, intra mode decision occupies the main part of RDO computation, so the speed of this method is not very remarkable.

By studying the entropy coding method in H.264/AVC, an accurate bit-rate estimation model is proposed in this chapter. The parameters in this model are self-adaptive to guarantee the precision of estimation. The distortion is also calculated in transform domain to avoid the MB reconstruction process. By using these methods, plenty of RDO computation is saved while keeping good performance.

In this chapter, an efficient intra mode selection algorithm for H.264 is proposed which is based on above two techniques: edge classification and rate-distortion estimation. There are three main contributions in this chapter:

1. An edge classification algorithm based on non-normalized Haar transform (NHT) is introduced, which is applied in H.264 intra mode selection.

2. An accurate bit-rate estimation model is proposed. Based on this model, a fast RDO method is proposed and lots of entropy coding computation is saved.

3. A fast intra mode RDO scheme is designed based on the above two techniques.

Experimental results show that our methods can greatly speed up the intra prediction process with nearly no degradation of coding performance.

The chapter is organized as follows: Section 2.2 gives a brief introduction of intra prediction and RDO in H.264. The proposed edge classification method is introduced in Section 2.3. In Section 2.4, a fast intra mode decision algorithm based on edge classification is presented. Section 2.5 describes our fast RDO algorithm, which is based on precise bit-rate estimation model. Experimental results are shown in Section 2.6 and this chapter is summarized in Section 2.7.

2.2 Overview of Intra Prediction and RDO in H.264

There are two kinds of intra prediction for luma block in H.264 baseline profile: I4MB and I16MB. In I4MB type, nine different modes are defined as illustrated in Figure 2.1 (b). Each 4×4 block is predicted from the spatial adjacent samples which have already been encoded and reconstructed. The 16 samples of the prediction block which

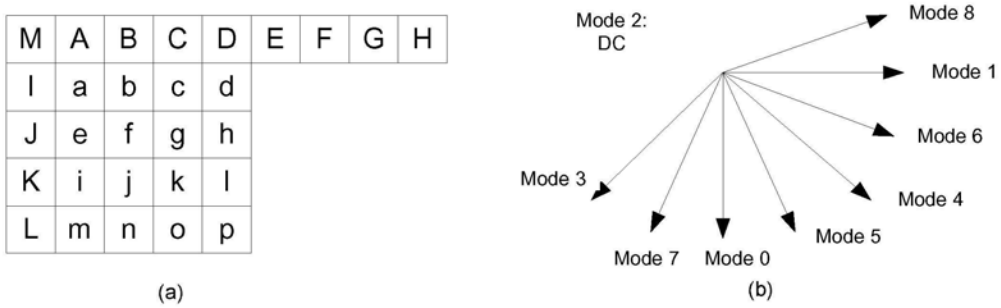

Figure 2.1: *(a) A 4 × 4 block with pixels (a-p) which are predicted by neighboring pixels (A-M) (b) 9 modes in 4 × 4 intra prediction*

are labeled as a-p are predicted by the neighboring pixels labeled as A-M as shown in Figure 2.1 (a). For example, mode 0 uses A, B, C and D to extrapolate the prediction block vertically. The other modes follow the similar method along their corresponding orientations. Mode 2 (DC mode) is a directionless mode in which all pixels are predicted by (A+B+C+D+I+J+K+L)/8. For the I16MB type, 33 reference pixels are used to generate a 16×16 prediction block, four modes are supported: vertical, horizontal, DC and the plane modes.

To choose the best macroblock mode, H.264/AVC encoder computes the RDcost for each possible mode and choose the mode which has the minimum RDcost as the best mode. In information theory, the RDO process can be described as looking for the minimum required rate R to achieve a given distortion D. RDO uses Lagrange Multiplier method to get the optimal solution. The cost function is shown as follows.

$$RDcost = SSE + \lambda_{mode} \times R \tag{2.1}$$

where SSE is the sum of squared error between the original block and the reconstructed block, λ_{mode} is the Lagrange multiplier, R represents the bit number consumed for coding this block. As shown in Figure 2.2, in order to compute the RDcost for each mode, the block needs to be encoded and decoded to get the bit-rate and the distortion, so a number of operations of forward/inverse transform and entropy coding are repeatedly performed. Hence the computational cost of RDO is very high.

In H.264/AVC, RDO_{off} option is also supported. The cost function of this mode is

$$RDcost = SAD + \lambda_{mode} \times R \tag{2.2}$$

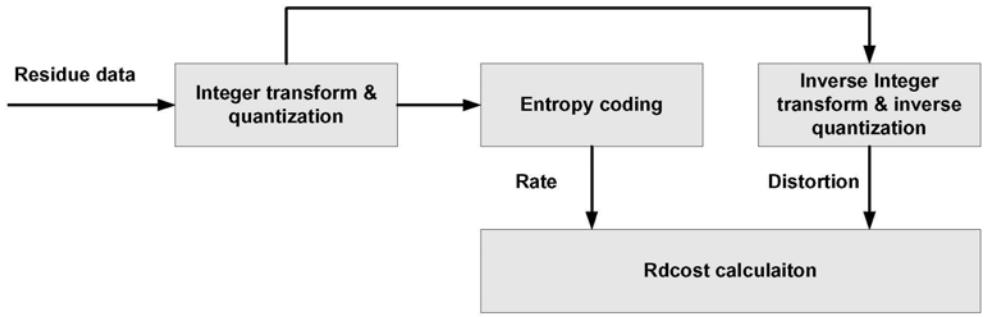

Figure 2.2: *RDO computation*

where SAD is the sum of absolute difference between the original block and the predicted block, R stands for the bits for coding mode type, motion vector and other side information. The entropy coding process is not needed in this mode, so the computation complexity is much lower than RDO_{on} mode, but at the same time, the coding performance is also much worse than RDO_{on} mode.

From Eqs. (2.1) and (2.2), it can be deduced that choosing I16MB means that fewer bits are used to signal the mode type and other information, but the residue may contain higher energy, especially in sequences with more details. On the other hand, choosing I4MB may give a lower energy residual after intra prediction but it needs to use more bits to signal the mode types and other information. So, how to select the best mode is very important for the performance of the H.264 codec.

2.3 Fast and Efficient Edge Classification Algorithm

In [Li and Ngan, 2006], a fast and efficient method was proposed to classify the edge blocks in terms of the NHT coefficients. Unlike the DCT, the NHT not only employs the fast multi-resolution structure, but also the Walsh-like transform. Because only addition and subtraction operations are involved in this transform, the computational complexity is greatly reduced when computing the NHT coefficients.

Let $x(n)$, $n = 0, \cdots, N-1$,with N even, denote a sequence of integers. The NHT can be represented by two sequences, the approximate $l(n)$ and detailed coefficients $h(n)$, defined as follows:

$$\begin{aligned} l(n) &= x(2n) + x(2n+1) \quad n = 0, \cdots, N/2-1 \\ h(n) &= x(2n) - x(2n+1) \quad n = 0, \cdots, N/2-1 \end{aligned} \tag{2.3}$$

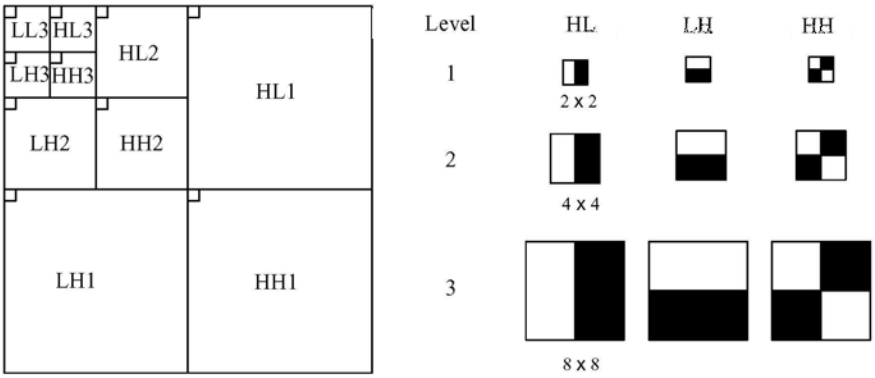

Figure 2.3: *(a) The multi-resolution representation of 2-D NHT. (b) Block features at different levels*

From Eq. (2.3), it can be found that NHT has the most efficient computational efficiency compared with other discrete transforms since the whole computation only requires a small number of additions and subtractions due to its function coefficients with only +1 and -1.

The 2-D NHT is performed by applying the transformation Eq. (2.3) sequentially to the rows and columns of the image. The corresponding hierarchical pyramid structure is shown in Figure 2.3 (a). It is noted that the same transformations are only applied to the reduced resolution LL sub-band to form the hierarchical pyramid. The coefficients at the λ_{th} level correspond to the $2^{\lambda} \times 2^{\lambda}$ pixel blocks due to the constant decomposition structure given in Eq. (2.3). Furthermore, for different subbands, i.e., LL, LH, HL, and HH, difference information corresponding to the original image can be observed. As shown in Figure 2.3(b), the same subbands at different levels, such as LH1, LH2, LH3,$\cdots$, will represent the grey level variation in similar way. The coefficients in LH subband denote the grey level change in the vertical direction, while the HL and HH represent the change in horizontal and diagonal directions respectively. LL is the DC component.

Let C denotes a $2^{\lambda} \times 2^{\lambda}$ block. B_0, B_1, B_2 and B_3 denote intensity sum of four non-overlap sub-blocks in a given block C, which are shown in Figure 2.4. Then, the

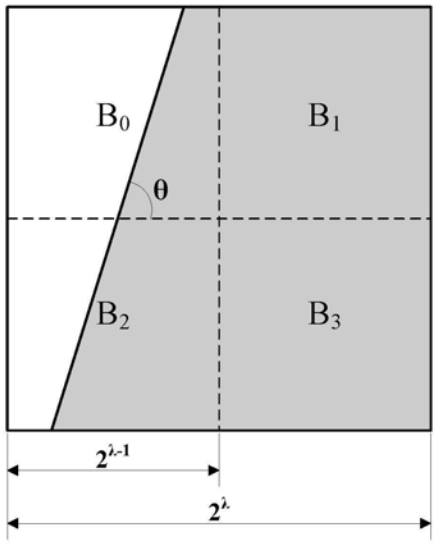

Figure 2.4: *Possible edge orientation in the given block*

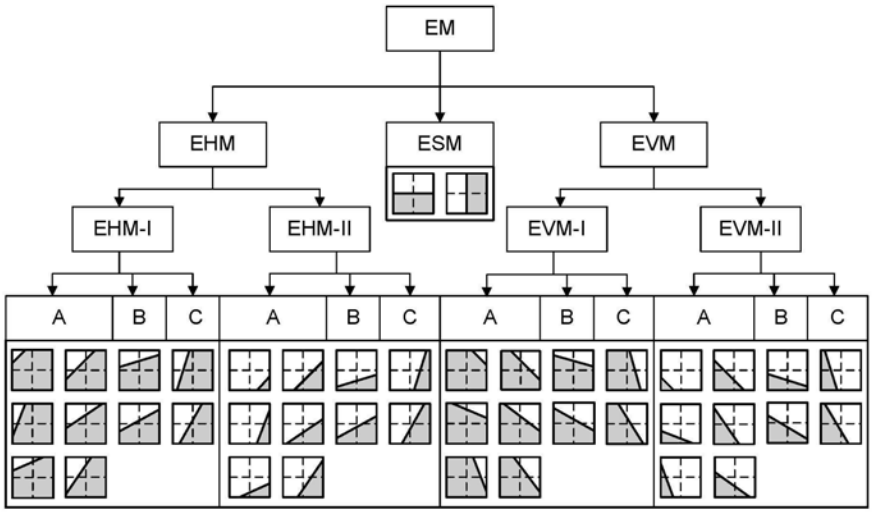

Figure 2.5: *The framework of the edge classification method*

four NHT coefficients at the λ_{th} level can be written as

$$\begin{aligned} LL &= B_0 + B_1 + B_2 + B_3 \\ LH &= B_0 + B_1 - B_2 - B_3 \\ HL &= B_0 - B_1 + B_2 - B_3 \\ HH &= B_0 - B_1 - B_2 + B_3 \end{aligned} \tag{2.4}$$

Based on three NHT AC coefficients and some simple geometrical knowledge, about 42 block edge models can be derived, which are shown in Figure 2.5. In other words,

according to some criteria described in [Li and Ngan, 2006], accurate edge information for a given block is obtained by only using three NHT AC coefficients at the highest level. Detailed classification process is described in [Li and Ngan, 2006].

2.4 Proposed Fast Intra Mode Decision Algorithm

As mentioned in the previous section, the edge model for each image block is identified by using the edge orientation feature. However, blocks with very small intensity changes could also be classified as the homogeneous block because of the weak edge strength. To identify blocks with small intensity change, the contrast information is used in the decision function, which is defined as

$$\begin{aligned} F &= f_{LH} + f_{HL} + f_{HH} \\ f_{LH} &= |LH/(LH + LL)| \\ f_{HL} &= |HL/(HL + LL)| \\ f_{HH} &= |HH/(HH + LL)| \end{aligned} \tag{2.5}$$

The block is classified as a homogeneous block when the decision function F is smaller than a certain threshold Th. In addition, each NHT coefficient is also set as zero if less than 0.6 Th.

Based on a lot of experiments, a threshold within the interval of 0.05 to 0.1 can provide a good classification result. Based on the classification results, corresponding intra modes are chosen as the candidates modes according to orientation of edge, then unnecessary computation will be saved.

2.4.1 I4MB Prediction Modes

For I4MB prediction modes, the three AC NHT coefficients LH, HL and HH are computed for a given 4×4 block first, then the edge of the block is identified. If the block is homogeneous, mode 2 (DC mode) is chosen as the candidate mode. If there exits a vertical or horizontal edge, mode 0 or mode 1 will be chosen for the RDO calculation, respectively. If the block contains a edge belonging to other edge modes according to the edge direction, corresponding modes can be selected as candidate modes. It should

Prediction Modes	0	1	8, 3, 7	5, 4, 6
Edge Model	ESM-2	ESM-1	EHM-I$A_{1,1}$ EHM-I$A_{x,2}$ EHM-II$A_{1,1}$ EHM-II$A_{x,2}$	EVM-I$A_{1,1}$ EVM-I$A_{x,2}$ EVM-II$A_{1,1}$ EVM-II$A_{x,2}$
Prediction Modes	0, 5, 4	4, 6, 1	3, 7, 0	1, 8, 3
Edge Model	EVM-I$A_{3,1}$ EVM-IC EVM-II$A_{3,1}$ EVM-IIC	EVM-I$A_{2,1}$ EVM-IB EVM-II$A_{2,1}$ EVM-IIB	EHM-I$A_{2,1}$ EHM-IC EHM-II$A_{2,1}$ EHM-IIC	EHM-I$A_{3,1}$ EHM-IB EHM-II$A_{3,1}$ EHM-IIB

Table 2.1: *The relationship between the prediction modes and the edge modes for* 4×4 *luma block*

be noted that only nine intra modes are defined in H.264 for a given 4×4 luma block, which are much less than 42 edge modes defined in our edge classification method. Table 2.1 gives the relationship between the nine prediction modes and the edge modes. The subscripts in this table denote the model position in Figure 2.5. For example, EHM-I$A_{x,2}$ refers to the three models at the second column for the case of EHM-IA shown in Figure 2.5. From this table, it could be found that at most three intra prediction modes are chosen for one detected edge. For example, if an edge belonging to EHM-IC model is detected for given block as shown in Figure 2.4 (edge orientation angle θ is within the range $(\pi/4\ \pi/2)$) , mode 0, mode 3 and mode 7 are chosen as the candidates according to Table 2.1.

An additional mode named mostProbableMode also needs to be checked, which is described in detail in H.264 specifications [ITU-T and ISO/IEC, 2005] and produced by the spatially adjacent blocks. There are two reasons to choose this mode. Firstly, if the correlation between adjacent blocks is very high, this mode has higher probability to be the best mode. Secondly, if the best mode for given block equals to the mostProbableMode, less bits are used to encode the mode index, so this mode is apt to be chosen as the best mode.

In summary, at least one mode, and at most four modes (three are from edge detection, one is from mostProbableMode) are chosen as the mode candidates to perform the RDO computation, instead of the original nine modes.

2.4.2 I16MB Prediction Modes

I16MB type produces the prediction block for the whole macroblock. It is more suited to encode the smooth area of the picture. On the other hand, I4MB type can get better result for coding the area with more details. Based on this observation, some criteria can be set to skip the unnecessary I16MB computation.

The proposed edge classification method is performed for the given 16×16 macroblock to check whether it is homogeneous. If this macroblock is homogeneous, I16MB type will be checked. Otherwise, this type will be skipped and computational cost will be saved. The procedure for checking the homogeneity of the macroblock is described as follows. First the three AC NHT coefficients are computed for the given macroblock. If any one of the following three conditions is satisfied, this macroblock will be considered as homogeneous.

a) HL, LH, HH are all zeros. It means the block is totally homogeneous.

b) HL and HH are zeros, but LH is not zero. It means that the block is homogeneous horizontally.

c) LH and HH are zeros, but HL is not zero. It means the block is vertically homogeneous.

Experimental results show that there is a very high probability that I16MB type is skipped. Especially for the sequence with many details, this ratio is more than 80%.

2.4.3 I8MB Prediction Modes

In H.264 baseline profile, I8MB is the type of intra prediction modes for chroma component. Totally four modes are supported, they are DC, vertical, horizontal and plane mode. Same as the previous method, first the edge direction is obtained from the U and V components based on the NHT coefficients. Then the mode candidates are decided according to the edge direction. If the edge directions from the two components are same, only one mode candidate is chosen; otherwise, two modes candidates need to be checked. The DC mode also needs to be checked in order to maintain the coding performance. Thus, two or three modes will be checked instead of the original four modes.

Table 2.2 shows the number of the candidate modes for each block size.

Block size	Original #	Min # for our method	Max # for our method
4×4	9	1	4
16×16	4	0	4
8×8	4	2	3

Table 2.2: *Number of candidate modes*

2.5 Proposed Fast RDO Algorithm

Although RDO improves the coding performance greatly, the computation complexity is too high to be acceptable for real-time applications. It is necessary to design a fast RDO algorithm to reduce the computational cost of this module. In this section, a fast RDO approach is proposed, which is based on the following three techniques.

2.5.1 Precise Bit-rate Estimation Model

In order to predict the bit-rate accurately, the components of bits for coding one block should be analyzed . Total bits for coding one block can be expressed by the following formula.

$$R_{total} = R_{coef} + R_{header} + R_{motion} \tag{2.6}$$

where R_{header} stands for the bits used for coding header information, such as mode type, coded block pattern (CBP). R_{motion} is the bit number of motion information, including motion vector, reference frame index, etc. The bit number of these two parts can be obtained through look-up tables easily.

R_{coef} represents the bits used for coding quantized coefficients. It is not practical to get the bits of this part by using look-up table, since the entropy coding method in H.264/AVC is much more complex. Through our study of the entropy coding algorithm in H.264/AVC, it is observed that the consumed bits for coding the quantized coefficients are related to three factors: N (the number of nonzero quantized transform coefficients), Z (summation of run-before), and E (summation of absolute value of total quantized coefficients).

In order to investigate the relationship between bit-rate and the quantized coefficients, some models was reported in recent literatures. He et al. [He et al., 2001; He and Mitra, 2002; He et al., 2002] proposed a ρ-domain rate model, which described the linear relationship between rate and the percentage of zero quantized DCT coefficients.

This model achieves very good results in some previous compression standards, such as H.263, MPEG-2. However, the entropy coding method adopted in H.264 is much more complex. As discussed before, it could be found that the ρ-domain model is only related to the factor N (the number of nonzero quantized transform coefficients), but ignores the other two factors Z and E, which also play the important roles in entropy coding. Therefore, ρ-domain model could not predict the bits encoded by H.264 accurately. Tu et al. [Tu et al., 2006] improved He's model. In this model, the predicted bits are equal to the linear combination of N and E. Thus, the better results can be obtained. In order to get a more accurate model to estimate the bit-rate, all the above three factors are considered in our model and the bits for coding quantized coefficients can be predicted by the following equation:

$$R_{coef} = \alpha \times N + \beta \times Z + \gamma \times E \tag{2.7}$$

where α, β and γ are three parameters of this function. In order to be adaptive to the features of different video sequences, or different frames in video, parameters α, β and γ can be computed and updated by using linear regression method.

For the $(n+1)_{th}$ macroblock, α, β and γ are obtained by solving following equation arrays:

$$\vec{R_n} = \alpha \times \vec{N_n} + \beta \times \vec{Z_n} + \gamma \times \vec{E_n} \tag{2.8}$$

where vector $\vec{R_n}$ contains n elements which are actual bits for encoding previous n macroblocks. $\vec{N_n}$, $\vec{Z_n}$ and $\vec{E_n}$ also record the actual values of N, E and Z in previous n macroblocks, respectively. It could be observed that there are three unknowns and n equations, so least squares (LS) method is applied to solve it. Since the derivation and result are very tedious and complex, the abbreviated expression of the result is shown as Eq. (2.9):

$$\alpha = T_\alpha / F \quad \beta = T_\beta / F \quad \gamma = T_\gamma / F \tag{2.9}$$

where:

$$\begin{aligned} T_\alpha &= S_{nr}(S_{zz}S_{ee} - S_{ze}^2) - S_{zr}(S_{ee}S_{nz} - S_{ne}S_{ze}) + S_{er}(S_{nz}S_{ze} - S_{zz}S_{ne}) \\ T_\beta &= S_{nr}(S_{ne}S_{ze} - S_{ee}S_{nz}) - S_{zr}(S_{nn}S_{ee} - S_{ne}^2) + S_{er}(S_{nz}S_{ne} - S_{nn}S_{ze}) \\ T_\gamma &= S_{nr}(S_{nz}S_{ze} - S_{zz}S_{ne}) - S_{zr}(S_{ne}S_{ze} - S_{nn}S_{ze}) + S_{er}(S_{nn}S_{zz} - S_{nz}^2) \\ F &= S_{nn}S_{zz}S_{ee} - S_{nn}S_{ze}^2 - S_{zz}S_{ne}^2 - S_{ee}S_{nz}^2 + 2S_{nz}S_{ne}S_{ze} \end{aligned} \tag{2.10}$$

In Eq. (2.10):

$$\begin{aligned} S_{nr} &= \sum_{k=1}^{n} N_k R_k \quad S_{zr} = \sum_{k=1}^{n} Z_k R_k \quad S_{er} = \sum_{k=1}^{n} E_k R_k \\ S_{nn} &= \sum_{k=1}^{n} N_k N_k \quad S_{zz} = \sum_{k=1}^{n} Z_k Z_k \quad S_{ee} = \sum_{k=1}^{n} E_k E_k \\ S_{nz} &= \sum_{k=1}^{n} N_k Z_k \quad S_{ze} = \sum_{k=1}^{n} Z_k E_k \quad S_{ne} = \sum_{k=1}^{n} N_k E_k \end{aligned} \tag{2.11}$$

where n stands for the number of encoded macroblocks in past, N_k, Z_k, E_k are the values of the three corresponding components of the k_{th} MB, R_k is the real consumed bits for coding the quantized coefficients of the k_{th} MB.

Although the above regression function looks very complex, in fact, the $(k+1)_{th}$ group of the parameters (α ,β ,γ) can be obtained easily by the k_{th} group of the parameters, so little extra computation cost is introduced in this regression process.

In Figure 2.6 (test sequence is Foreman.QCIF) x-axis is the actual bits, and y-axis is the estimated bits. The beeline is 45 degree line. It can be observed that the most of dots lie on the narrow range around the fitted line. It means that our bits estimation model can predict actual coding bits accurately, saving much computation in entropy coding.

2.5.2 Fast Intra Mode RDO Method

H.264 allows intra modes to be used in the inter frames. Therefore, when performing mode selection, not only inter modes but also intra modes need to be checked. The 4×4 block can be regarded as the counter unit of RDO operation because transformation, quantization and entropy coding are all use 4×4 as basic unit. Since there are 16 4×4 blocks for a given macroblock and eight inter modes defined in H.264 as illustrated in

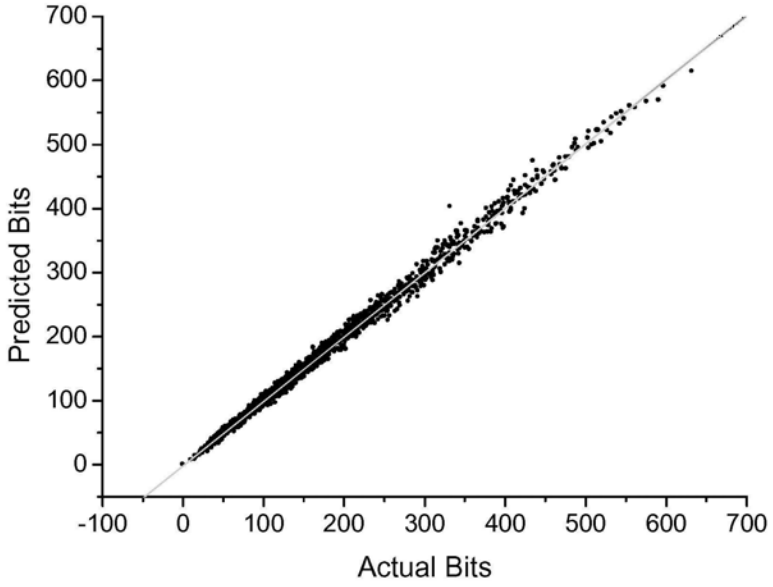

Figure 2.6: *Correlation between actual bits and predicted bits*

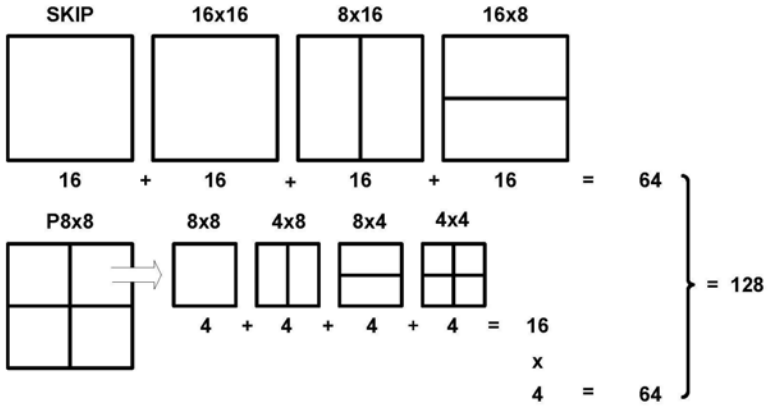

Figure 2.7: *The total number of RDO operations in inter mode decision [Jeon, 2003]*

Figure 2.7, it is found that a total of 128 RDO operations are needed for inter mode decision [Jeon, 2003].

For intra mode decision, H.264/AVC combines luma and chroma components together to perform RDO operation since the symbol named CBP is related to not only the chroma intra mode type, but also the luma intra mode type. In order to obtain the accurate RDcost where bits for coding CBP is considered, the following procedure is performed in RDO: under a fixed chroma mode, all the luma modes are checked to calculate the RDcost. Then chroma mode is changed and all luma modes are checked again until the best combination of chroma and luma mode is obtained which makes the RDcost minimum. Because there are four chroma modes, as shown in Figure 2.8,

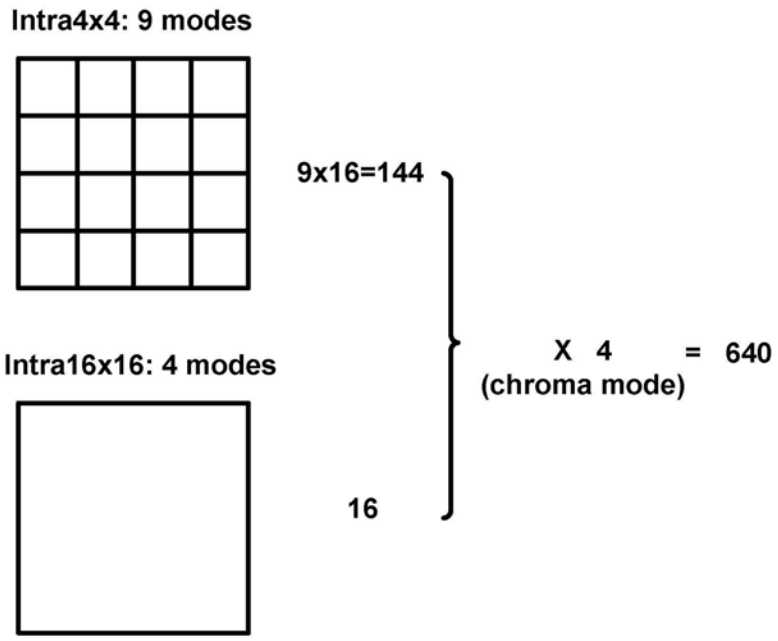

Figure 2.8: *The total number of RDO operations in intra mode decision [Jeon, 2003]*

the total number of RDO operations for intra mode decision is 640 [Jeon, 2003], which can be calculated by Eq. (2.12) and is much higher than that of inter mode decision. Because RDO is a very time-consuming operation and is more complex than motion estimation/compensation (ME/MC) in inter frame coding when fast ME algorithm is applied, for one macroblock, the complexity of intra mode selection is higher than that of the inter mode decision when RDO is turned on. It is necessary to reduce the intra mode RDO computation.

$$N_{RDO} = M8 \times (16 \times M4 + 16) \tag{2.12}$$

In Eq. (2.12), N_{RDO} is the total number of RDO operations. $M8$ and $M4$ are 4 and 9, respectively, which stand for the mode number of chroma and luma I4MB. In H.264 codec reference software JM [HHI, 2005], mode selection for I16MB type is performed by following steps. First, Hardamard transform is performed on the residue signals of four I16MB modes and the mode with minimum summation of Hardamard coefficients as the best I16MB mode is chosen. Then RDcost is calculated for the best I16MB mode by using Eq. (2.1) to compared with I4MB modes and inter modes, so only 16 units of RDO operation are performed for I16MB mode selection in the JM codec.

Since the bits for encoding CBP is very few, luma and chroma components can

be considered approximatively independent when performing intra mode selection. By trading off accuracy versus complexity in intra mode decision, RDO computation for the chroma and luma components can be performed separately to choose the best chroma mode and luma mode. In this case the total number of computing RDcost is calculated by Eq. (2.13), and only 164 RDO operations are needed instead of original 640 RDO operations.

$$N_{RDO} = M8 + (16 \times M4 + 16) \tag{2.13}$$

Based on Table 2.2, in the best scenario, our algorithm needs to perform $2 + (16 \times 1 + 0) = 18$ RDO operations. In the worst case, this number is $3 + (16 \times 4 + 16) = 83$. Compared with 640 RDO operations in the original H.264 intra prediction method, the proposed algorithm can reduce the computational complexity significantly. Experimental results also show that the reduction of RDO operations has limited impact of coding performance. It needs to be clarified that this fast intra RDO method differs from other traditional fast intra mode selection algorithms because it does not reduce the mode types which will be checked. So, this fast intra mode RDO scheme can be integrated with other fast mode selection algorithms to achieve much faster encoding speed.

2.5.3 Distortion Computation in Transform Domain

In original H.264 algorithm, the reconstructed frame need to be generated in order to calculate the distortion between the original frame and the reconstructed frame. Because the integer transform in H.264/AVC is an orthogonal transform and distortion is measured by SSE (sum of squared error) in RDO, according to conservation of energy, the quantization error in transform domain must be equal to the distortion in spatial domain. The quantization error can be calculated in transform domain. Then, dequantization and inverse ICT computation can be avoided. This idea was also described in literatures [Chen and He, 2004] and [Tu et al., 2006].

Since ICT uses left and right shift operations to replace multiplication and division, the complexity of ICT is very low compared with DCT. This is also the reason that ICT is adopted in H.264. The ICT only occupies a very small proportion in RDO operations, so distortion computation in transform domain contributes to our fast RDO

algorithm slightly. The speedup ratio of proposed algorithm is mainly from the former two techniques.

2.6 Experimental results

The proposed algorithms were integrated within the H.264 software JM10.1 [HHI, 2005]. Here our two methods were tested. Proposed method I only adopted the fast intra mode decision algorithm (as discussed in Section 2.4). Proposed method II showed the hybrid effect of the proposed fast mode decision and the fast RDO algorithms. Another fast intra decision method proposed by Pan [Pan and Lin, 2004; Pan and Lin, 2005] was also implemented. All of the above methods were compared with the full modes search method in JM10.1 with RDO_{on}. In order to evaluate the proposed fast RDO algorithm, JM10.1 with RDO_{off} was also compared. The system hardware was a PC with 2.8GHz Intel P4 CPU and 1Gb memory. Several test sequences with 150 frames were chosen and covered high, mild and low motion, as well as QCIF and CIF resolution. Quantization parameters were set as 28, 32, 36 and 40, entropy coding adopted CAVLC.

In order to evaluate coding efficiency, the coding performance is measured in terms of ΔT_{total}, ΔT_{RDO}, ΔPSNR and ΔBR. ΔT_{total} is the saving ratio of total encoding time defined in Eq. (2.14), when RDO is turned on. ΔT_{RDO} stands for the coding time saving ratio of the RDO module and is calculated by Eq. (2.15). The values of ΔT_{total} and ΔT_{RDO} shown in the following tables are average time saving ratios in the tests with upper four QP values. ΔPSNR and ΔBR denote the differences of PSNR and bit-rate which are calculated by using the RD-curves fitting method in [Bjontegaard, 2001].

$$\Delta T_{total} = \frac{T_{proposed} - T_{Jm}}{T_{Jm}} \times 100\% \tag{2.14}$$

$$\Delta T_{RDO} = -\frac{T_{proposed} - T_{RDO_{on}}}{T_{RDO_{off}} - T_{RDO_{on}}} \times 100\% \tag{2.15}$$

2.6.1 All Intra Frames Mode

In this experiment, all 150 frames were coded as intra frames. Tables 2.3 and 2.4 show the coding performance compared with JM 10.1 with RDO_{on} and the encoding

Sequence		RDO_{Off}		Pan's method		Proposed I		Proposed II	
		ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]
Q	Foreman	-0.41	6.61	-0.15	2.47	-0.15	2.48	-0.17	2.77
C	Container	-0.54	8.46	-0.20	2.97	-0.21	3.16	-0.21	3.22
I	Carphone	-0.40	5.73	-0.22	3.16	-0.38	5.24	-0.39	5.39
F	MissA	-0.37	6.77	-0.35	6.18	-0.22	3.80	-0.22	3.63
	Bus	-0.49	7.42	-0.22	3.45	-0.21	3.23	-0.24	3.60
	Paris	-0.44	5.87	-0.20	2.64	-0.20	2.73	-0.21	2.82
C	Hallmonitor	-0.56	8.92	-0.30	4.64	-0.16	2.51	-0.21	3.22
I	Mobile	-0.51	6.19	-0.23	2.81	-0.23	2.86	-0.24	2.91
F	MotherDr	-0.46	8.99	-0.33	6.27	-0.21	3.17	-0.28	5.21
	News	-0.48	6.77	-0.29	3.97	-0.28	3.53	-0.32	4.28
	Football	-0.40	7.94	-0.16	3.02	-0.15	2.88	-0.21	4.08
	Coastguard	-0.41	7.34	-0.16	3.02	-0.17	2.56	-0.19	3.42
Average		-0.46	7.25	-0.23	3.72	-0.21	3.18	-0.24	3.71

Table 2.3: *Coding performance compared with JM10.1* RDO_{On} *(All intra frame)*

Sequence		RDO_{Off}	Pan's method	Proposed I	Proposed II	
		ΔT_{total} [%]	ΔT_{total} [%]	ΔT_{total} [%]	ΔT_{total} [%]	ΔT_{RDO} [%]
Q	Foreman	-87.05	-51.68	-59.54	-83.56	95.99
C	Container	-86.85	-50.42	-60.45	-83.32	95.94
I	Carphone	-86.68	-50.22	-60.31	-83.14	95.92
F	MissA	-85.10	-47.56	-58.35	-80.79	94.94
	Bus	-88.77	-54.42	-60.86	-85.81	96.67
	Paris	-89.14	-57.09	-62.31	-85.57	96.00
C	Hallmonitor	-86.60	-51.28	-60.43	-83.38	96.28
I	Mobile	-89.29	-53.63	-60.52	-85.24	95.46
F	MotherDr	-85.81	-51.45	-59.28	-82.81	96.50
	News	-86.75	-53.09	-60.25	-83.25	95.97
	Football	-86.58	-50.66	-59.97	-83.11	95.99
	Coastguard	-87.78	-51.85	-60.24	-84.77	96.57
Average		-87.20	-51.95	-60.21	-83.73	96.02

Table 2.4: *Coding time saving ratio compared with JM10.1* RDO_{On} *(All intra frame)*

time saving ratio, respectively. The first column contains the results of the JM 10.1 with RDO_{off}. When RDO is turned off in JM10.1, the speed is much faster, but the coding performance is degraded greatly. The PSNR decreases about 0.46 dB on average and the bit-rate increases about 7.25%. The third column shows the results of proposed method I. It could be observed that the performance of our method is better than Pan's method and the speed is also faster. The reason is that our proposed edge

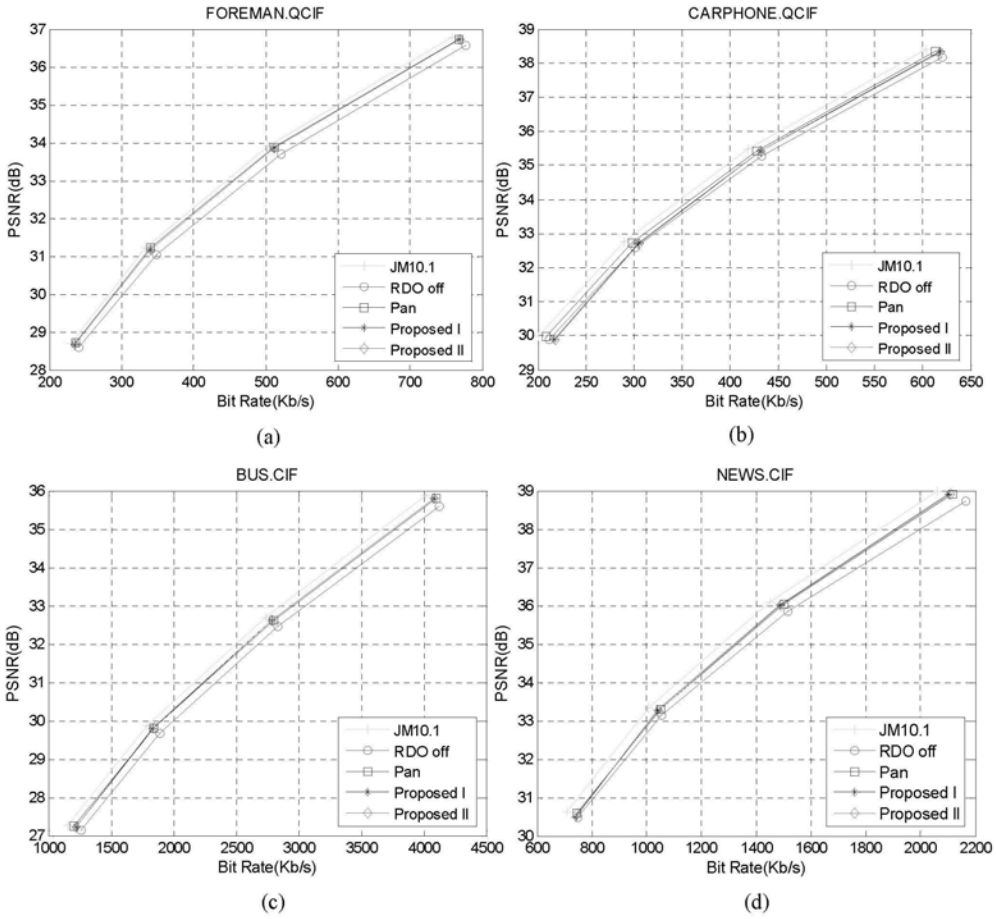

Figure 2.9: *Rate-distortion curves (for all I-frames): (a) Foreman(QCIF). (b) Carphone(QCIF). (c) Bus(CIF). (d) News(CIF)*

detection algorithm is based on Haar transform and only a small number of addition and subtraction operations are involved. Therefore, the extra computational cost is very low. However, In Pan's method, Sobel operator is operated on each pixel and the edge direction histograms also need to be calculated, so the complexity is a little higher. The last column tabulates the results of proposed method II, which is the combination of the fast intra mode decision method and the fast RDO algorithm. Our proposed method II can save over 80% coding time and about 96% computational cost of the RDO part. Compared with JM10.1 and Pan's method, our algorithm is much faster. At the same time average increment of bitrate is about 3.7% or equivalently the loss of PSNR is about 0.24 dB. The loss is very slight and can almost be ignored. Figure 2.9 plots the R-D curves of several sequences. The performance difference between our method and JM10.1 is negligible.

2.6.2 IPPP...Mode

In H.264 inter frame coding, not only the inter modes, but also the intra modes have to be checked. For a given macroblock, the mode with the minimum cost will be chosen as the best mode to encode the macroblock. Although most of the macroblocks are finally encoded by the inter modes, the intra mode selection is still a necessary step in the inter frame coding. As discussed in Section 2.5.2 , when RDO is turned on, the RDO operations in intra mode selection will consume more computational costs than those in the inter mode decision, so our fast intra decision and fast RDO algorithms are also effective when coding inter frames. By applying our proposed algorithm, fewer intra modes will be calculated and a number of unnecessary RDO operations are reduced. In this experiment, for a total of 150 frames, the first frame was I-frame and all other following frames were coded as P-frame. The reference frame number was 1. The Simplified UMHexagonS [Yi et al., 2005] was choosen as the fast motion estimation algorithm.

The experimental results are shown in Tables 2.5 and 2.6. The speed and performance of our proposed method I are better than those of Pan's method. Our method II can reduce 60% of the total computation time compared with JM10.1 on average with almost the same bitrate and PSNR. The results are also consistent with our previous analysis. Figure 2.10 shows the R-D curves of the four test sequences. It could be observed that the curves of ours and JM10.1 method almost overlap each other. This implies that the performance of proposed fast algorithms is nearly the same as that of JM10.1.

2.7 Summary

This chapter introduces a fast intra mode decision algorithm based on fast edge detection method and rate-distortion estimation. There are three main contributions in this chapter: The first one is to design an edge classification method which is based on non-normalized Haar transform (NHT) and it is applied in fast intra mode selection scheme. The second one is to propose an accurate bit-rate estimation model, and based on this model, much entropy coding computation is saved. The third one is to design a fast intra mode RDO scheme based on upper two techniques. Verified by the fast, mild, slow motion sequences, our method could reduce the computational complexity by

Sequence		RDO_{Off}		Pan's method		Proposed I		Proposed II	
		ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]
Q	Foreman	-0.39	9.12	0.01	-0.12	0.00	0.07	-0.02	0.37
C	Container	-0.67	16.55	-0.03	0.79	-0.02	0.55	-0.14	2.95
I	Carphone	-0.11	2.59	-0.01	0.16	0.03	-0.77	-0.03	0.71
F	MissA	-0.89	19.78	-0.03	1.06	-0.08	1.55	-0.16	3.49
	Bus	-0.34	7.82	-0.00	-0.01	-0.01	0.34	-0.04	0.84
	Paris	-0.39	8.66	0.01	-0.16	-0.01	0.17	-0.08	1.82
C	Hallmonitor	-0.54	16.66	-0.01	0.30	0.00	-0.07	-0.09	2.66
I	Mobile	-0.24	6.29	0.01	-0.16	0.00	0.05	-0.00	0.07
F	MotherDr	-0.25	6.85	-0.00	0.11	-0.01	0.29	-0.12	3.00
	News	-0.44	9.08	-0.07	1.46	-0.11	2.25	-0.17	3.30
	Football	-0.50	11.81	-0.04	0.95	-0.04	1.06	-0.17	3.76
	Coastguard	-0.33	10.51	0.00	-0.05	-0.02	0.79	-0.03	0.89
Average		-0.42	10.48	-0.02	0.36	-0.02	0.52	-0.08	1.98

Table 2.5: *Coding performance compared with JM10.1* RDO_{On} *(IPPPP....mode)*

Sequence		RDO_{Off}	Pan's method	Proposed I	Proposed II	
		ΔT_{total} [%]	ΔT_{total} [%]	ΔT_{total} [%]	ΔT_{total} [%]	ΔT_{RDO} [%]
Q	Foreman	-77.69	-40.48	-46.83	-62.53	80.49
C	Container	-80.55	-42.87	-48.76	-64.57	80.17
I	Carphone	-78.58	-40.65	-45.13	-62.51	79.55
F	MissA	-79.14	-39.15	-44.83	-60.94	77.00
	Bus	-76.01	-41.06	-48.04	-63.97	84.16
	Paris	-80.18	-44.23	-50.93	-66.52	82.96
C	Hallmonitor	-79.61	-42.70	-50.72	-65.10	81.77
I	Mobile	-79.23	-42.93	-49.68	-66.71	84.20
F	MotherDr	-78.24	-41.17	-49.22	-62.72	80.16
	News	-79.21	-43.56	-50.22	-64.73	81.72
	Football	-74.20	-38.01	-44.23	-60.01	80.88
	Coastguard	-75.46	-40.34	-47.25	-63.20	83.75
Average		-78.18	-41.43	-47.99	-63.63	81.39

Table 2.6: *Coding time saving ratio compared with JM10.1* RDO_{On} *(IPPPP....mode)*

choosing the best mode judiciously. Average time reduction is over 80% in all I-frame sequences and over 60% in IPPP sequences. Moreover, our algorithm can maintain the video quality without significant bit-rate loss. The fast algorithm can be applied to real-time implementation of H.264 encoder in low-power applications of video coding.

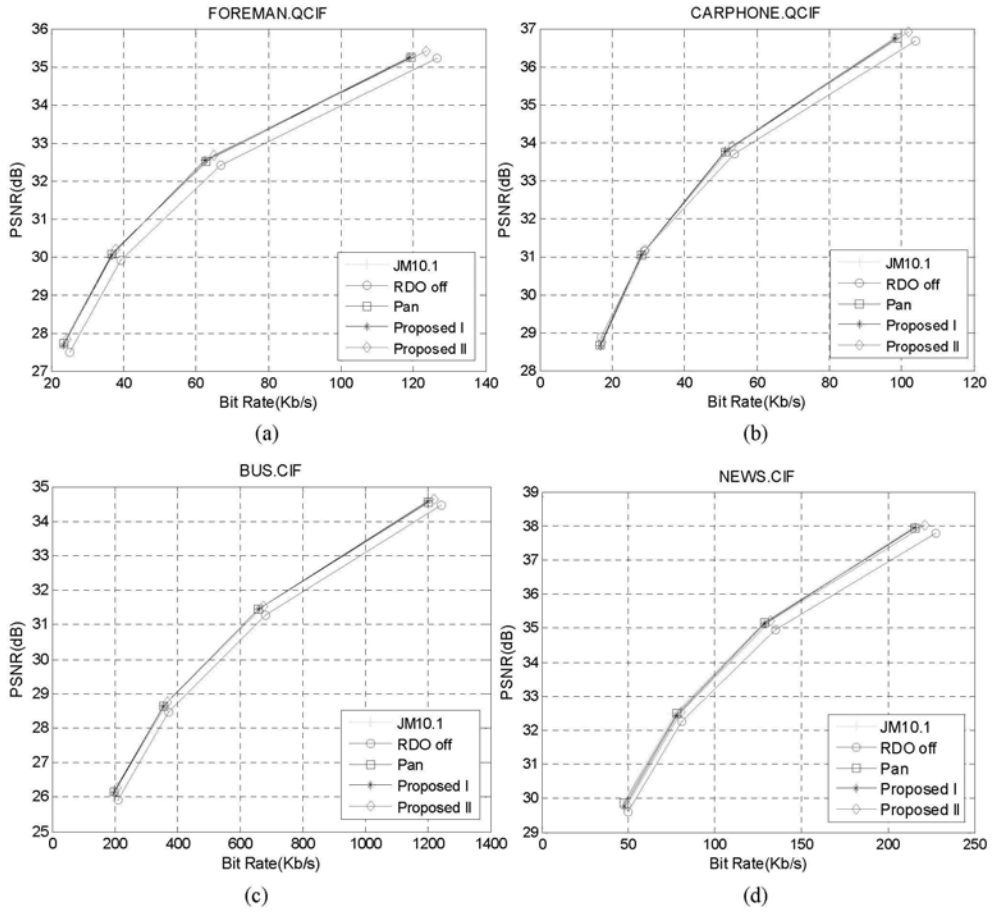

Figure 2.10: *Rate-dsistortion curves (IPPP... mode): (a) Foreman(QCIF). (b) Carphone(QCIF). (c) Bus(CIF). (d) News(CIF)*

Part of the work in this chapter was presented at *ISCAS2007*, entitled "An Efficient Intra Mode Selection Algorithm For H.264 Based On Fast Edge Classification", and *ICASSP2007*, entitled " A Fast Rate-Distortion Optimization Algorithm For H.264/AVC", respectively. Part of the work in this chapter was published in *IEEE Transactions on Circuits and System for Video Technology* as a regular paper, entitled "Fast and Efficient Method for Block Edge Classification and Its Application in H.264/AVC Video Coding". All the work in this chapter was published in *Signal Processing: Image Communication* as a full length article, entitled "An Efficient Intra Mode Selection Algorithm for H.264 Based on Edge Classification and Rate-Distortion Estimation".

Chapter 3

Fast Inter Mode Decision Algorithm for H.264

3.1 Introduction

As introduced in Section 2.1, in order to achieve outstanding coding performance, H.264 adopts many advanced techniques, including variable block size modes in inter frame coding. There are seven different block size modes defined in H.264. Moreover, Pskip mode and the intra modes are also allowed in inter frame coding. In previous video coding standards such as H.263, MPEG-1, MPEG-2, only 1 or 2 block sizes are allowed. So it is obvious that the seven different block size modes consume a lot of computational loads of the H.264 encoder. It is very important to propose an efficient fast inter mode decision method for optimizing H.264 codec. Ahmad's method [Ahmad et al., 2004] uses the neighboring MBs' modes to predict the current mode, but because the mode mismatch influences the bit rate and PSNR greatly, the performance is not good. Zhang's method [Zhang and Zhang, 2004] is based on the results of different resolution motion estimation. It can save computational cost of sub-pixel search significantly and has potential to be improved further. In this chapter, we propose a fast mode selection algorithm which is achieved by applying the Pskip mode early detection, early termination and mode prediction techniques. In order to keep the coding quality, we use the post-search technique. We also use intra mode skip detection to avoid unnecessary intra mode calculation. From the experimental results, we can see that this method can greatly speed up the motion estimation process with nearly no degradation of coding performance.

The rest of the chapter is organized as follows: Section 3.2 introduces the multiple inter modes in H.264. The proposed fast inter mode decision algorithm is described in Section 3.3. Experimental results are given in Section 3.4. The last section summarizes the chapter.

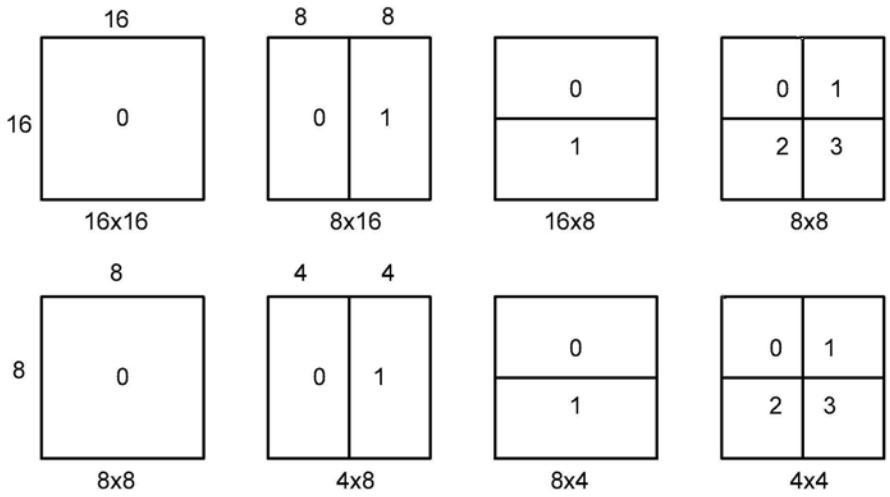

Figure 3.1: *Variable block size in H.264*

3.2 Variable Block Size in H.264

In H.264 inter frame coding, seven different block size modes are defined as shown in Figure 3.1. They are divided into two types: MB (macroblock) partitions consist of 16×16, 16×8, 8×16 block sizes; sub-MB partitions (named P8×8 modes) consist of 8×8, 8×4, 4×8, 4×4 block sizes. Furthermore, In P frame, H.264 has three other modes: Pskip, Intra4×4, Intra16×16. Pskip mode is a direct copy of the previous frame. Intra4×4 and Intra16×16 are intra coding modes to encode the current MB.

The best mode selected and used for coding must give the minimal cost. When RDO is turned off, the cost function is given as follows.

$$J(m, \lambda_{motion}) = SAD(s, c(m)) + \lambda_{motion} \cdot R(m\text{-}p) \tag{3.1}$$

where, $m = (m_x, m_y)^T$ is the current MV (motion vector), $p = (p_x, p_y)^T$ is the predicted MV. $R(m - p)$ represents the bits used to encode the MV information. $SAD(s, c(m))$ stands for the sum of absolute difference between current MB and reference MB. λ_{motion} is the Lagrange multiplier.

From Eq. (3.1), we know that choosing a larger partition size means that fewer bits are used to signal the MVs and other information, but the residue may contain higher energy, especially in sequences with more details. On the other hand, choosing a smaller partition size may give a low energy residual after motion estimation but it needs to use more bits to signal the MVs and other information. So, how to select the

best mode is very important to get the best performance of the H.264 codec.

3.3 Proposed Fast Inter Mode Decision Algorithm

3.3.1 Pskip Mode Early Detection Based on Transform Domain

Pskip is a kind of special mode. In the case of skipped MB, the residue will not be encoded and the current MB is just a direct copy of previous frame. In many video sequences, some MBs are decided as Pskip mode after the computation of all modes because they are background or have low-motion. In H.264, if a MB is in Pskip mode, it must satisfy the following conditions: (1) The best mode size is 16×16. (2) The motion vector is equal to the predicted motion vector. (3) All the DCT coefficients after quantization are zeros. (4) The reference frame must be the neighboring frame before the current frame.

If we can set a relaxed criterion to check the Pskip mode first without any priori knowledge, a lot of computation cost of the remaining modes will be saved. This will increase the speed of the encoder significantly. Because the Pskip mode does not encode the residue, an inaccurate decision will influence the coding performance adversely. However, the original conditions of Pskip detection are too rigid to be satisfied. It is very important to design a proper rule not only to have more MBs to be in Pskip mode but also not to affect the coding performance too much. Since one of the important conditions to check for Pskip mode is that all the DCT coefficients are zeroes, setting a threshold in pixel domain cannot reflect the distribution of the transform coefficients well. It is reasonable to use the thresholding technique in transform domain.

In the 4×4 DCT transform, there is a high chance that very sparse coefficients exist in 8×8 or 16×16 block and they are very small (level=1). Encoding them will consume a lot of bits but does not have many contributions to the R-D improvement. So we propose a method to judge whether the coefficients are small enough and can be

	Bus	Foreman	Container
Matching ratio	65.80%	82.30%	98.19%

Table 3.1: *Mode matching ratio between current MB and neighbor MBs.*

discarded. The method is described as follows.

$$\begin{aligned}
&\text{If level} \quad > 1 \,\text{or DC coefficients is non-zero coe_cost} = 9 \\
&\text{If level} \quad = 1 \quad \text{coe_cost} = \begin{cases} 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{cases} \textit{when run_before} = \begin{cases} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{cases} \\
&\text{other} \qquad\quad \text{coe_cost} = 0
\end{aligned}$$

$$luma\_cost = \sum_{0}^{15}\sum_{0}^{15} coe\_cost \tag{3.2}$$

$$chroma\_cost = \sum_{0}^{3}\sum_{0}^{15} coe\_cost \tag{3.3}$$

where level is the absolute value of the quantized coefficient, run_before is the number of consecutive zero levels before a non-zero level. Luma_cost is the sum of all coe_costs in MB, chroma_cost is the sum of all coe_costs in 8×8 Cr or Cb block.

For a MB, if the luma_cost is less than 6, chroma_cost of Cr and Cb are all less than 9, this MB will be decided as Pskip mode.

3.3.2 Mode Prediction Method

Since there is high correlation between the adjacent MBs, using information of neighboring blocks to predict the information of current block is effective. This method has been applied to motion vector prediction widely, such as median prediction for MVs which has been accepted as a part of the H.264 standard. We could apply this technique to the mode prediction to increase the encoder speed. Table 3.1 shows the mode matching ratio between the current MB and its adjacent MBs. we can see there is high

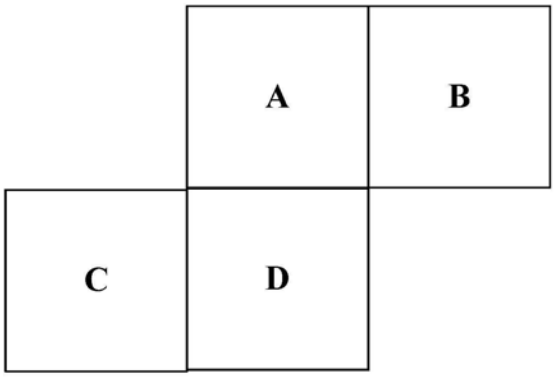

Figure 3.2: *Mode prediction method. A: upper MB; B: up-right MB; C: left MB; D: co-located MB in reference frame*

correlation between current MB and its neighboring MBs, especially in low motion sequence (Container sequence). As shown in Figure 3.2, the adjacent MBs include the MBs located on the left, above-right and above the current MB in the current frame, and the co-located MB in the previous frame.

From the discussion above, it is feasible to predict the mode of current MB using the modes of adjacent MBs. The formulation is shown below.

$$CurrentMode = Min\_cost(modeA, modeB, modeC, modeD) \tag{3.4}$$

3.3.3 The Early Termination Technique

We also apply the early termination technique on mode checking by employing an adaptive threshold. If the minimal cost of any mode is less than a threshold, this mode is considered to be good enough to be the best mode and mode searching will be stopped. The threshold T_1 is set in the following equation:

$$T_1 = Min(mincost\_A, mincost\_B, mincost\_C, mincost\_D) \tag{3.5}$$

3.3.4 The Post-search Technique

Because the mode can influence the final coding quality greatly, the precision of the mode decision is important. In some cases, the neighbor MB cannot reflect the current MB's mode. For example, if the current MB contains fast motion and reference MBs are all background, using these MBs for prediction is not correct. From Table 3.1, we can find that in the high motion sequences, the correlation between the adjacent MBs

	MB level	P8×8	Intra
Flower	63.74%	34.81%	1.45%
Foreman	91.93%	6.87%	1.20%
Grandma	98.35%	0.65%	1.00%

Table 3.2: *Ratio of each level of mode being best mode.*

	Foreman	Salsman	Container	Bus
Intra	3454.0	3545.4	3651.5	4527.8
Inter	1630.6	1339.3	1614.5	2230.1

Table 3.3: *Average cost of best intra mode and best inter mode.*

is not high. If we solely use the mode prediction technique, the loss of PSNR can reach $0.5 \sim 1$dB [Khan et al., 2004] and the bit-rate will increase to about 20% [Ahmad et al., 2004]. In order to solve this problem, we set another threshold called post-threshold, if the minimum cost of the candidate modes is larger than this threshold, we will search the all the modes which have not been searched. The threshold is:

$$T_2 = Max(mincost\_A, mincost\_B, mincost\_C, mincost\_D) \tag{3.6}$$

3.3.5 Intra Mode Skip Detection

In H.264, intra modes are allowed in inter frame. From Table 3.2, we can find that the probability that a MB is coded as intra mode in inter frame is very low, but in H.264 mode decision algorithm, all intra16×16 and intra4×4 modes will be checked after checking inter modes. If we can judge whether intra mode detection is necessary for current MB in advance, a lot of computational cost will be reduced. Table 3.3 shows the mean value of best inter mode cost and best intra mode cost respectively. It is obvious that intra mode cost usually is much larger than inter mode cost for a MB. Motivated by this observation, an intra mode selection algorithm is proposed. We only need to check the intra16×16_DC mode which is the simplest intra mode. If the cost of best inter mode is less than half of the cost of intra16×16_DC mode, intra mode detection will be skipped. Otherwise, intra mode checking is still needed.

3.3.6 The Approach of The Fast Mode Decision

Based on the five considerations above, our fast mode decision algorithm is described as follows. Figure 3.3 shows the flowchart of the proposed scheme.

Step 1: Detect the pskip mode as described in Section 3.3.1. If true, end the search.

Step 2: Mode prediction

1. When MB lies in the first row or the first column of the frame, all seven modes will be searched.

2. If the MB is located in the right edge of the frame, we only use the modes of the MBs in positions A, C and D (in Figure 3.2) as the current MB's predicted mode will be calculated.

3. If the MB lies in the middle, we use the modes of the MBs in positions A, B, C and D as the candidate modes of the current MB.

Step 3: Searching

Set a threshold as in Eq. (3.5). If the cost of the current mode is less than the threshold, the search will be stopped and the current mode is selected as the best mode of the current MB. Otherwise, search all candidate modes, compare their costs and decide the best mode. If the MB lies in the left and upper edge of the frame, the threshold is set as zero. If MB lies in the right edge, we only use MB A, C and D to calculate threshold.

Step 4: Post-search

Set a post-threshold as in Eq. (3.6). If the cost of the best inter mode in step 3 is larger than the post-threshold, search all the modes which have not been searched and select the best inter mode.

Step 5: Intra-skip detection

Compute the cost of intra16×16 DC mode. If cost of best inter mode is less than half of cost of intra16×16 DC mode, skip all intra detection and the best inter mode is the best mode for current MB. Otherwise, check all the intra modes to compare with best inter mode and decide the final mode.

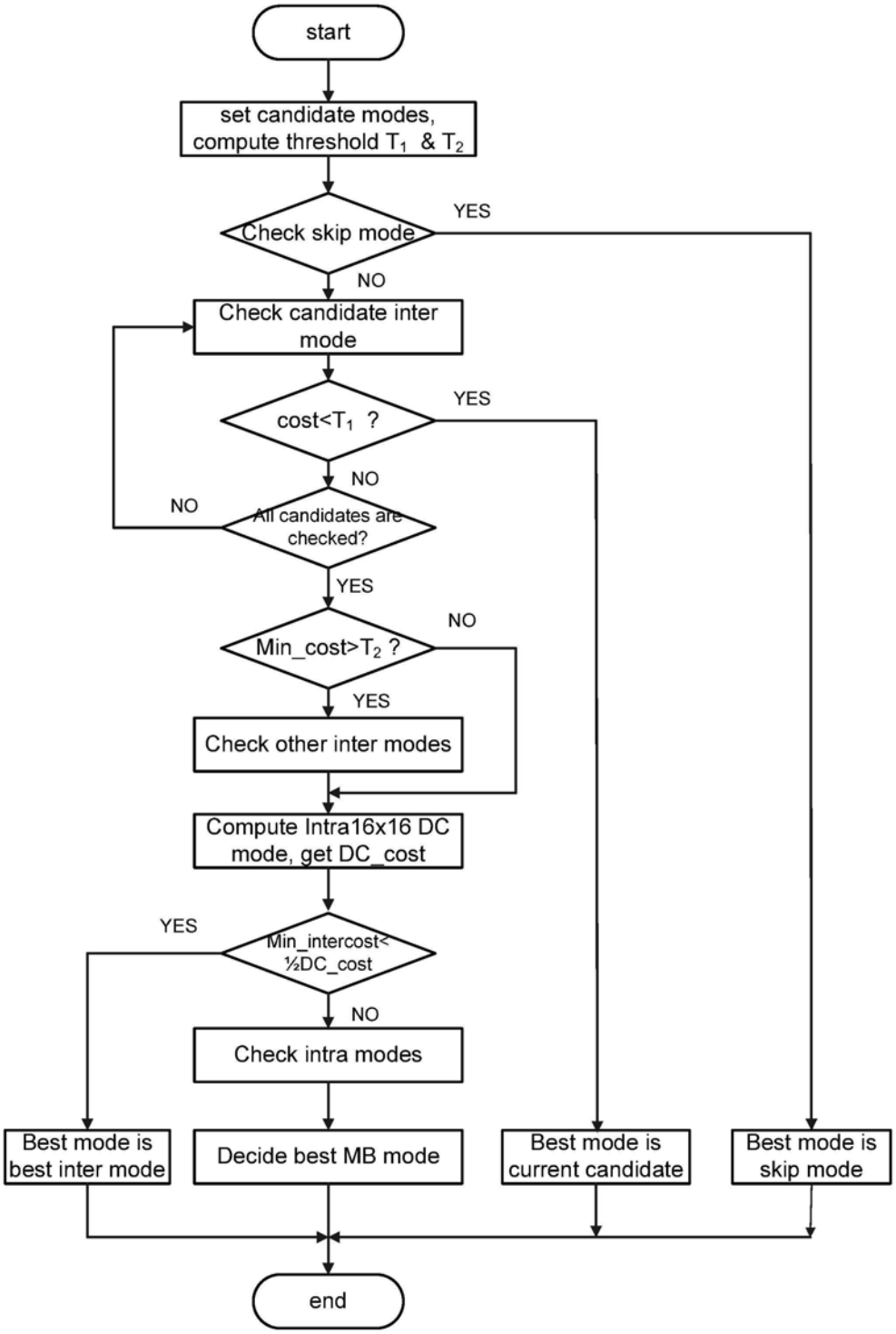

Figure 3.3: *Flowchart of the proposed inter mode decision scheme*

Format	Sequence	Proposed		Andy's		Ahmad's	
		ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]
QCIF	Foreman	2.433	-0.1053	3.927	-0.1677	10.627	-0.4494
	Salesman	1.438	-0.0766	2.061	-0.1058	15.579	-0.7454
	MissA	-5.176	0.2233	3.389	-0.1341	7.703	-0.3147
CIF	Paris	1.565	-0.0848	1.930	-0.1039	16.615	-0.8614
	Container	2.635	-0.0860	0.987	-0.0326	5.538	-0.1803
	Bus	2.099	-0.1088	1.439	-0.0767	12.915	-0.6456
	Waterfall	1.279	-0.0506	2.391	-0.0942	6.204	-0.2415
Average		0.896	-0.0413	2.304	-0.1021	10.740	-0.4912

Table 3.4: *Comparison of bit-rate and PSNR*

Formant	Sequence	Proposed	Andy's	Ahmad's
		ΔT [%]	ΔT [%]	ΔT [%]
QCIF	foreman	-39.933	-16.588	-24.713
	salesman	-58.259	-17.560	-33.911
	MissA	-57.807	-19.153	-33.279
CIF	Paris	-50.910	-22.237	-36.899
	container	-56.710	-15.828	-37.754
	bus	-31.714	-19.106	-23.613
	waterfall	-44.226	-24.317	-38.247
Average		-48.508	-19.256	-32.631

Table 3.5: *Comparison of the coding time saving ratio*

3.4 Experimental Results

Our proposed algorithm is integrated within the H.264 software JM10.1 [HHI, 2005], and it is compared with the full modes search method. The other two fast mode decision methods proposed by Andy [Yu et al., 2006] and Ahmad [Ahmad et al., 2004] are implemented to compare with our method. Our system hardware is a PC with 2.8GHz Intel P4 CPU and 1Gb memory. Seven test sequences with 150 frames are chosen and cover high, middle and low motion. The other settings are as follows: coding structure is IPPP$\cdots$, frame rate is 30 fps, search range is $\pm$16, use CAVLC entropy coding, use 1 reference frame, QP = 24, 28, 32, 36; choose Simplified UMHexagonS [Yi et al., 2005] as fast motion estimation algorithm.

Tables 3.4 and 3.5 show the coding performance of these three algorithms against JM10.1. ΔBR and ΔPSNR are two measures of encoding quality and are calculated according to upper 4 QPs [Bjontegaard, 2001]. ΔT indicates the average processing time

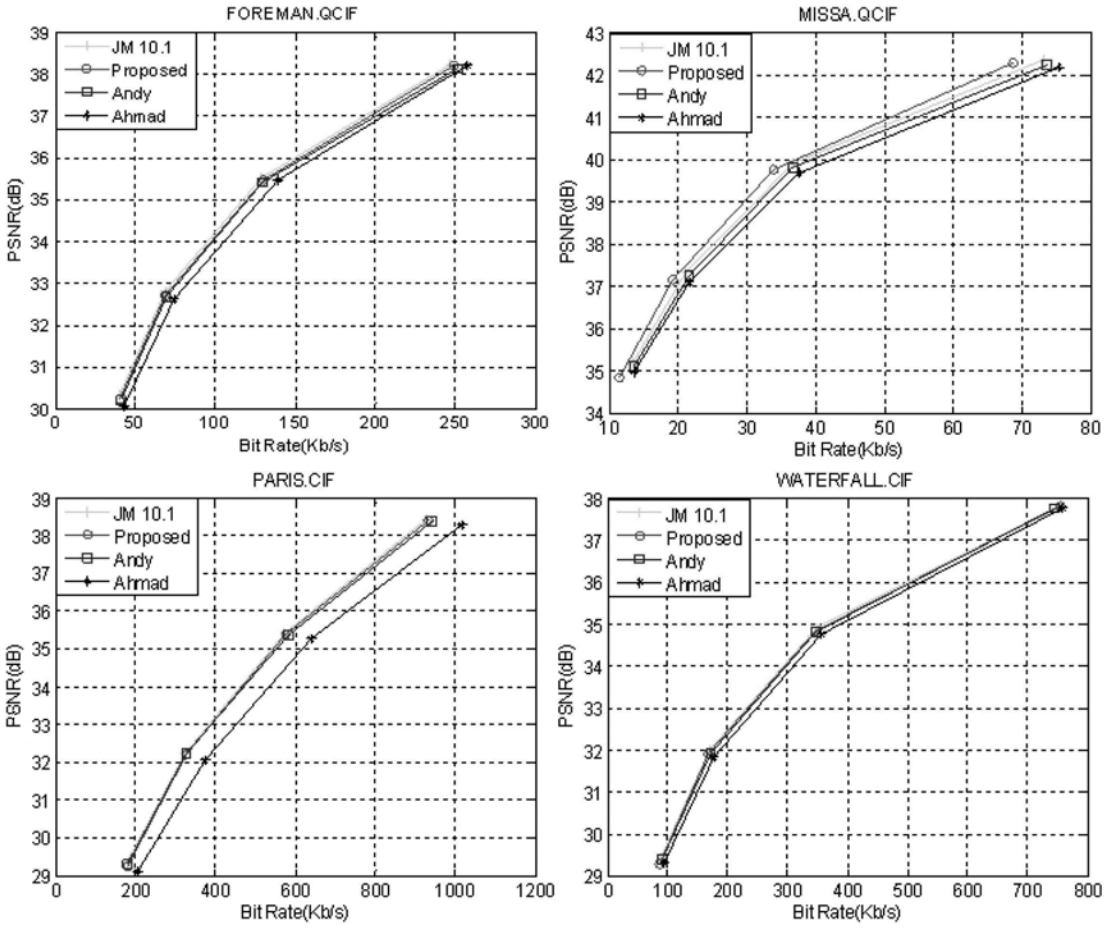

Figure 3.4*: Rate-distortion curves*

saving ratio for the upper QPs. Figure 3.4 plots the R-D curves of several sequences.

From the results, we find that our algorithm did not degrade the coding efficiency and picture quality. We can see that our method saves about 30%∼60% of coding time compared with JM10.1 algorithm and is faster than the other two fast mode decision algorithms. For some low motion sequences, such as MissA and Container, the correlation among the adjacent MBs is high and there is a higher probability to use the candidate modes as the best mode. Furthermore, low motion results in more zero DCT coefficients and plenty of Pskip MBs. So our proposed algorithm will be more effective. On the contrary, for the large motion sequences, such as Bus, our method will consume more time. Our method will be faster with the increase of the quantization parameter. That is because with large QPs, more DCT coefficients are quantized to zero and more MBs will be decided to be Pskip mode. For the sequence MissA, coding performance

is even better than JM10.1. That is because plenty of MBs are coded by pskip mode in this sequence and lots of bits are saved. Ahmad's method can keep bit-rates better in low motion sequence but worse in large motion sequence. The reason is that in high motion case the correlation between the adjacent MBs is not high, so mode prediction is not effective. This is also consistent with previous theoretical analysis.

3.5 Summary

In this chapter, we proposed a fast inter mode decision algorithm based on the Pskip early detection, mode prediction, early termination, post-search techniques and intra mode skip detection. Verified by the fast, middle, slow motion sequences, our method could reduce the computational complexity by choosing the best mode judiciously. Moreover, our algorithm can maintain the video quality without significant bit-rate loss. It is helpful for the real-time implementation of the H.264 encoder and useful for the low-power applications of video coding.

All the work in this chapter was presented at *APCCAS2006*, entitled "A Fast Macroblock Mode Decision Algorithm for H.264".

Chapter 4

Implementation of H.264 on Mobile Device

4.1 Introduction

As introduced in the previous sections, H.264/AVC [ITU-T and ISO/IEC, 2005] is the latest video coding standard of the ITU-T VCEG and the ISO/IEC MPEG. To achieve a highly efficient multimedia processing platform by combining H.264 and a steady embedded processor is significant in engineering and has a lot of market values. We choose Intel PXA27x Processor as our embedded processor, on which we implemented the H.264 standard. With the 624MHz processing power, this embedded processor is most suitable to overcome the complexity and computational requirements of H.264.

Due to the high complexity of H.264 and the constraints of computation resource of the mobile device, it is a challenging work to achieve a real time codec on embedded processor. We used many techniques and proposed some efficient algorithms to reduce the computational cost of the encoder and decoder, including system optimization, algorithm optimization and instruction optimization etc. For algorithm optimization, a fast integer pixel and sub-pixel motion estimation algorithm as well as an efficient inter mode decision method are proposed. They can speed up encoding time drastically. For instruction optimization part, we adequately used the abundant WMMX (wireless MMX) instructions [Intel, 2002] which are provided by Intel PXA27x Processor and are very suitable for multimedia processing. By using these optimization methods, a real time H.264 encoder is implemented on the mobile device. Since there is a decoding module in H.264 encoder, it is very easy to achieve a real-time decoder based on our optimized encoder. Finally, a real-time H.264 based video conferencing system is implemented on the mobile device.

The rest of chapter is organized as follows: Section 4.2 introduces our hardware platform briefly. Section 4.3 shows our implementation work. Section 4.4 and Section

4.5 discuss the algorithm and instruction optimization methods in details. The experimental results are shown in Section 4.6. The implemented H.264 based real-time mobile video conferencing system is introduced in Section 4.7. Finally, the conclusion is presented in Section 4.8.

4.2 Overview of PXA27x Processor

The simulation platform of our project is a HP C4700 PDA (personal digital assistant), whose CPU is an Intel PXA27x embedded processor. The Intel PXA27x processors are the first Intel XScale technology-based processors to include Intel Wireless MMX technology [Intel, 2002] to enable high-performance multimedia acceleration with an industry proven instruction set.

PXA27x processors have powerful computational capability and are very suitable for multimedia processing. PXA27x's core frequency is up to 624Mhz. Intel also added many new technologies to the PXA27x family such as:

- Wireless MMX: 43 new SIMD (single instruction, multiple data) instructions containing the full MMX instruction set and the integer instructions from Intel's SSE (streaming SIMD extensions) instruction set along with some instructions unique to the XScale. Wireless MMX provides 16 extra 64-bit registers that can be treated as an array of two 32-bit words, four 16-bit halfwords or eight 8-bit bytes. The XScale core can then perform up to eight adds or four MACs (multiply-accumulate) in parallel in a single cycle. This capability is used to boost speed in decoding and encoding of multimedia and in playing games.
- Additional peripherals, such as a USB-Host interface and a camera interface.
- Internal 256 KB SRAM to reduce power consumption and latency.

These features create the opportunity for parallel processing, so PXA27x processors is very suitable for image/video processing.

4.3 Implementation and Optimization of H.264 on Embedded System

In order to implement and optimize H.264 encoder on embedded system, we should carry out the following steps: profile selection, code porting and the code optimization,

which are explained below.

- Profile Selection

  H.264 standard supplies several profiles for different applications, including baseline profile, main profile, extended profile and high profile. We chose the baseline profile for implementation for two reasons. Firstly, the objective of baseline profile is to serve low bit-rate video communications and this is consistent with our application. Secondly, other profiles contain many very complex optional techniques which are not suitable to be implemented on embedded device, and the techniques adopted by the baseline profile are good enough to satisfy our requirement for implementation of high performance multimedia communication system.

- Code Porting

  Code porting is to port the existing C code on the embedded system and let it run on the PDA. The main problems in this step are memory allocation, modification of some functions or syntaxes which are not supported by embedded system.

- Code Optimization

  Since the code just ported on PDA has much redundancy and executive efficiency is too low to satisfy our requirement for real time application, it is necessary to optimize the code to increase the encoding speed and efficiency. The optimization of embedded program has following three levels:

  First is the program level optimization. It is a kind of global optimization method for the program. The main methods include: One method is to use the optimization option provided by the compiler to optimize the code. Another way is to modify the program structure and reduce the logical branch because it will destroy the program pipeline and influence the executive efficiency of the code.

  The second level is the algorithm level optimization. Based on the features of H.264 itself, we proposed some fast and efficient algorithms to improve the encoding speed to achieve our goals.

The third level is the instruction level optimization. PXA27x processor supports wireless MMX instruction sets and provides an abundance of multimedia processing instructions, for example, we can perform 8 additions, subtractions or SAD (sum of absolute difference) operations for 8-bit data in only one instruction. This is very useful and valuable for video processing.

Since H.264 encoding loop includes a decoding module, it is very easy to implement the decoder by using the optimized encoder. From what we have introduced above, the steps and basic methods of the implementation and optimization of H.264 on mobile device have been discussed. Next, we would like to discuss the algorithm level and instruction level optimization in more details.

4.4 Algorithm Optimization

By studying the algorithms of H.264, it can be found that motion estimation, sub-pixel search and mode selection consume about 70%∼80% of the computational cost of H.264 encoder, so it is necessary to optimize these parts. In our optimization work, fast inter/intra mode decision algorithms, fast integer/fraction pixel motion estimaiton algorithms are proposed and employed in our codec. These methods can increase the efficiency of the encoder and produce good image quality with almost the same bit rate. In the following parts, we will introduce these algorithms in detail.

4.4.1 Block Mode Decision

In H.264, 7 different block size modes are defined, from 16×16 to 4×4. The encoder chooses the mode with the minimum cost as the best mode to process the current MB. It is obvious that the 7 different block size modes in H.264 standard consume a lot of computational resources of the encoder. It is very important to design an efficient fast mode decision method for optimizing the H.264 codec.

In order to perform cost comparison when doing block size decision or motion estimation, several cost functions are defined by Eq. (4.1).

$$\begin{aligned} MEcost_{M\times N} &= Distortion + \lambda_{motion} \times R(MV) \\ MBcost &= Distortion + \lambda_{mode} \times R(MV, ModeType) \\ SubMBcost &= Distortion + \lambda_{mode} \times R(MV, ModeType) \end{aligned} \tag{4.1}$$

where, $M \times N$ stands for block size, which is from 16×16 to 4×4. MEcost is the cost of motion estimation. MBcost is the total MB cost of mode decision. SubMBcost is the cost of SubMB when performing mode decision. Distortion means the difference between original block and corresponding reference block. Here we use SAD as the distortion. λ_{motion} and λ_{mode} are the Lagrange multipliers for motion estimation and mode decision respectively. R represents the bits used to encode the MV, ModeType and other overhead information.

In order to speed up the speed of block size decision, several techniques are used in our encoder which will be described in detail as follows:

1) Pskip Mode Early Detection:

As introduced in Section 3.3.1, Pskip is a kind of special mode. In the case of skipped MB, the residue will not be encoded and the current MB is just considered as a direct copy from previous frame. If we can set a relaxed criterion to check the Pskip mode first without any priori knowledge, a lot of computation cost of the remaining modes will be saved. This will increase the speed of the encoder. Here we borrow the Pskip early detection method in Section 3.3.1. The detailed algorithm can be found in Section 3.3.1 or our previous publication [Wei and Ngan, 2006].

2) Block Size Selection:

After performing Pskip mode early detection, a lot of unnecessary block type checking calculation will be saved. But if the current MB cannot satisfy the criterion of skip mode, other block types should be checked.

The proposed block size decision algorithm is based on the top-down approach. If the MBcost, which is defined in Eq. (4.1), of 16×16 block size is smaller than that of 8×16 and 16×8, the MBcost of 8×8 block size will not be computed and 16×16 block size will be regarded as the best block size. Otherwise, the MBcost of 8×8 block size will be computed.

For a sub-MB, we do the same as 16×16, 8×16, and 16×8, i.e., if the SubMBcost of 8×8 block size is smaller than that of 4×8 and 8×4, the SubMBcost of 4×4 block size will not be computed. This process is applied for all of the four sub-MBs.

To achieve a faster speed, a early termination condition is added. First, $MBcost_{16\times16}$, $MBcost_{16\times8}$ and $MBcost_{8\times16}$ are computed, the minimum one is considered as the $MBcost_{best}$. Then, for 8×8 block size, the $SubMBcost_{idx}$ (idx is the sub-MB partition index) is computed based on Eq. (4.1), the best sub-MB partition size is selected amongst the 8×8, 8×4, 4×8 and 4×4 sub-MB partition sizes. After computing $SubMBcost_0$ and $SubMBcost_1$, if $(SubMBcost_0 + SubMBcost_1) >= 0.6 \times MBcost_{best}$, the remaining two SubMBcosts and the MBcost of 8×8 block size will not be computed and the current best block size will be considered as the final best block size. Otherwise, $SubMBcost_2$ and $SubMBcost_3$ will be computed and $MBcost_{8\times8}$ will be found. Then $MBcost_{8\times8}$ will be compared with the $MBcost_{best}$ and the one with a smaller value will be considered as the $MBcost_{best}$ and the corresponding block size will be regarded as the best block size for the current MB.

4.4.2 Intra Mode Decision

Intra mode prediction is an effective method to eliminate spatial redundancy by using adjacent pixels to predict current MB. Since there are many intra modes defined in H.264 standard, the computational cost of intra mode decision is very high and it is very necessary to propose a fast intra mode decision algorithm.

Because H.264 allows intra MB coding in inter frame, but for most MBs, intra cost is much larger than inter cost and these MBs should be coded by inter mode. So we can set a threshold to check inter mode cost, if inter mode cost is smaller than this threshold, we can consider inter mode is good enough and the intra modes are not needed to be checked any more. Therefore, a lot of computation can be reduced.

In our encoder, we carry out inter mode decision first, then obtain the minimum cost of the inter modes. We then compare the inter mode cost with the Th_{intra} defined in Eq. (4.2). If inter mode cost is smaller than this threshold, the intra mode checking will be skipped. Otherwise, intra mode decision is still needed.

$$Th_{intra} = \frac{\sum_i^N Intracost(i)}{N} \tag{4.2}$$

where, N is the total number of previous coded MBs which are coded as intra type. $Intracost$ is the cost of each intra MB. The threshold Th_{intra} equals to the average value of all previous coded intra MBs' costs.

4.4.3 Fast Motion Estimation Algorithm

Motion estimation is the most time-consuming module in video coding. A fast motion estimation algorithm [Wei and Zhang, 2004] is proposed by using prediction and early termination techniques. The steps of the algorithm are briefly described as follows.

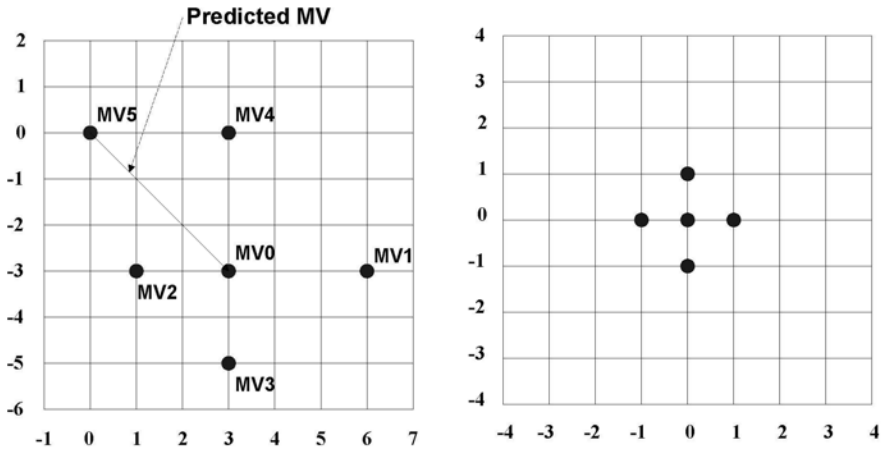

Figure 4.1: *Global search pattern and the refined local search pattern*

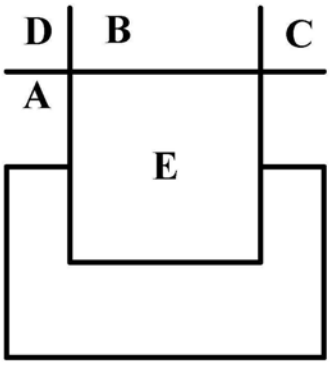

Figure 4.2: *Definition of the neighboring blocks (E is Current Block; A, B, C, D are neighboring blocks)*

Step 1: [Prediction] Calculate the predicted motion vector MV_0. The detailed prediction steps are described in the H.264 standard [ITU-T and ISO/IEC, 2005].

Step 2: [Initial Global Search] The search pattern is shown in Figure 4.1, which includes 6 searching candidates (A cross pattern $MV_0 \sim MV_4$ and the (0,0) point MV_5). Searching candidate vectors $MV_1 \sim MV_4$ are obtained by following equations.

$$\begin{aligned} \vec{MV}_1 &= \{max(MV_x), MV_{predicted-y}\} \\ \vec{MV}_2 &= \{min(MV_x), MV_{predicted-y}\} \\ \vec{MV}_3 &= \{MV_{predicted-x}, min(MV_y)\} \\ \vec{MV}_4 &= \{MV_{predicted-x}, max(MV_y)\} \end{aligned} \tag{4.3}$$

where MV_x, MV_y are the horizontal and vertical components of all the neighboring blocks' MVs (shown in Figure 4.2). $MV_{predicted-x}$ and $MV_{predicted-y}$ are the horizontal and vertical component of the predicted motion vector obtained from step 1.

The first candidate search vector to be evaluated is MV_0, a threshold T_1 is set to be $2 \times N_p + b$, where N_p is the number of pixels in the current block, b is a constant and is usually equal to 12$\sim$20. When the SAD is less than T_1, the search is stopped and MV_0 is the best motion vector. If the SAD is greater than T_1, compute the SAD of all other five points in Figure 4.1. Here a new threshold T_2 is set. If SAD is less than T_2, the search is stopped and this vector is the motion vector that we need. Otherwise, the search continues until the next step. T_2 is defined as:

$$T_2 = Min(MSAD_1, MSAD_2 \cdots MSAD_i \cdots MSAD_n, 3 \times N_p) + b \tag{4.4}$$

where $MSAD_i$ is the minimum distortion value of the neighboring blocks (shown in Figure 4.2) that have already be encoded. b is a constant.

Step 3: [Refined local search] Place the search center on the point which has the minimum SAD in the last step. The search pattern is shown in the right part of Figure 4.1. If the SAD is below T_2, search is stopped, otherwise, check the points until the center point of the diamond has the minimum SAD. This point is the minimum matching error (MME) point and the final motion vector is obtained.

After applying our algorithm, the average number of search points in integer pixel motion estimation is reduced to 4$\sim$5. If the search window is set as 32, full search needs to check 4225 points. The speed of our algorithm is about 1000 times of full

search. Compared with other classical ME (motion estimation) method, our algorithm also shows a superior performance in term of search speed. For example, the minimum search number is 13 in diamond search and 17 in new three step search.

4.4.4 Sub-pixel Motion Estimation

H.264 allows 1/4 pixel resolution motion estimation. After using the fast integer pixel motion estimation algorithm, the searching points are reduced greatly, but the number of search points in sub-pixel full search is still 16 in 1/4 resolution. Sub-pixel motion estimation consumes a lot of time of the H.264 encoder. An efficient fast sub-pixel search algorithm is necessary. Here, a fast sub-pixel motion estimation method is proposed and described as follows, which is an improvement of algorithm [Chen et al., 2002].

After finding the best integer motion vector, the searching center is shifted to the best integer matching point, then the sub-pixel motion estimation is carried out. The cost of three different candidate points will be computed. The coordinates of the three candidate points can be derived using the formulae defined in Eq. (4.5):

$$\begin{aligned} C_1 &= (0,0) \\ C_2 &= (MV_{predicted-x}\%4, MV_{predicted-y}\%4) \\ C_3 &= ((MV_{predicted-x} - MV_{integer-x})\%4, (MV_{predicted-y} - MV_{integer-y})\%4) \end{aligned} \quad (4.5)$$

where, $MV_{predicted-x}$ and $MV_{predicted-y}$ are the x and y component of the predicted MV which is defined in the H.264 standard and are in quarter-pixel units. $MV_{integer-x}$ and $MV_{integer-y}$ are the x and y component of the best integer motion vector selected through the fast integer ME algorithm which are multiplied by four for converting from integer-pixel units to quarter-pixel units. Here % indicates modulus operation.

The candidate points with the minimum cost will be considered as the center point for the following process. Here Th_{sub} is set to the cost of the candidate point C_1.

In the next step, the costs of four candidate points which are the neighbors of the current center point are computed based on small diamond search pattern as shown in Figure 4.3.

The candidate points with the minimum cost amongst the five candidate points

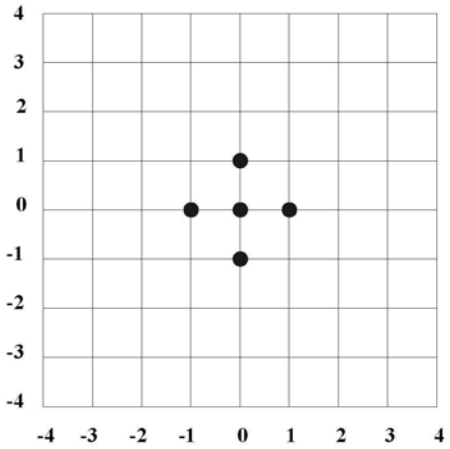

Figure 4.3: *Small diamond search pattern*

will be considered as the best candidate point at the current stage. If the best candidate point is not at the center of the search pattern, the best candidate point will be considered as the center point for the next stage of searching. The whole process will terminate only when the current best candidate point is at the center of the search pattern or any of the following conditions, which will be examined each time when the center point requires to be updated, is true, i.e.,

1. $MEcost < 0.6 \times Th_{sub}$
2. $MEcost < 0.2 \times Th_{intra}$

The current best candidate point will be considered as the best match.

4.4.5 Other Optimization Methods

H.264 Baseline Profile supports quarter-pixel ME. In our encoder, the sub-pixel will be generated only when it requires to be investigated. This can save the computational time consumed by the interpolation process. In addition, in order to derive a quarter-pixel, one or two half-pixel(s) must first be derived. Therefore, each time after a half-pixel is derived (when the corresponding half-pixel requires to be evaluated or a quarter-pixel, which must be derived from the corresponding half-pixels, requires to be evaluated), the half-pixel(s) derived will be stored for future use. This can greatly reduce the time consumed by the interpolation process and thus the sub-pixel ME process.

4.5 Instruction Optimization

After using algorithm level optimization, there is still room to improve the speed by employing the abundant multimedia instructions provided by Intel PXA27x embedded processor to optimize our codec further.

The Wireless MMX technology provided by Intel integrates the high performance MMX technology and the integer functions from SSE to the Intel XScale microarchitecture. Like MMX [Intel, 1999] technology and SSE [Intel, 2001], Wireless MMX technology utilizes 64-bit wide Single Instruction Multiple Data (SIMD) instructions which allow it to concurrently process up to eight 8-bit data in a single cycle.

Wireless MMX technology exploits the data parallelism presented in a large number of multimedia algorithms by executing the same operation on different data elements in parallel. This is accomplished by packing data elements into a single register and introducing new types of instruction to operate on packed data. All Wireless MMX technology data types are 64-bits wide. There are three packed data types defined in Wireless MMX technology. They are Packed Byte, Packed Half-word and Packed Word as shown in Figure 4.4

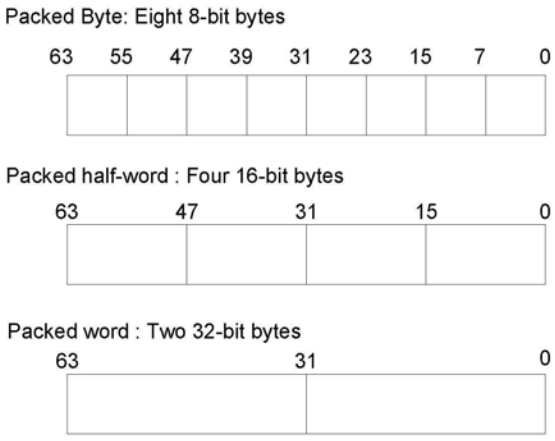

Figure 4.4: *Wireless MMX technology data types [Intel, 2002]*

Wireless MMX technology supplies a lot of new instructions which can operate on data elements in packed format. It means that here a single instruction operates on multiple data elements. Many multimedia algorithms execute the same set of operations on a large number of data elements. This creates an opportunity for parallel processing.

For example, when processing still images or video frames, the same operation is most often done over all of the pixels of the image or of the frame. Pixels are usually represented using 8-bit or 16-bit data elements. By executing these operations two, four, or eight at a time, Wireless MMX technology speeds up applications that exhibit data parallelism.

Wireless MMX provides sufficient parallel instructions, such as WADDB (perform 8 additions for 8-bit data pairs), WSUBB (perform 8 subtractions for 8-bit data pairs), WSADB (perform SAD calculation for 8 pairs of 8-bit data), WLDRB and WSTRB (load or store 8 8-bit data one time). These instructions are very useful and helpful for our implementation work.

We utilize the abundant multimedia instructions to rewrite some time-consuming functions by using assembly language. Involved modules include: ICT, motion estimation, sub-pixel search, interpolation etc. The effect is remarkable.

4.5.1 SAD Calculation

Motion estimation is the most time-consuming module in video coding. In H.264, since variable block sizes and 1/4 resolution motion estimation techniques are employed, the complexity of motion estimation module increases significantly. SAD (sum of absolute difference) is used as distortion measure criterion when performing motion estimation. The formula is shown in Eq. (4.6)

$$SAD = \sum_{i=1}^{M} \sum_{j=1}^{N} |O_{i,j} - R_{i+mv_x, j+mv_y}| \tag{4.6}$$

where, M and N are width and height of the block, $O_{i,j}$ is the intensity value of the original pixel in the i_{th} column and the j_{th} row. $R_{i+mv_x, j+mv_y}$ stands for the intensity value of the reference pixel in the i_{th} column and the j_{th} row. (mv_x, mv_y) is the motion vector. From this formula, we can see that SAD calculation needs many additions, subtractions and absolute value operations.

In Wireless MMX instruction set, instruction *WSAD* can perform the sum of absolute differences of two 8-bit data vectors (each vector has 8 data), and accumulates the results. So, one instruction can calculate SAD value of 8 pixels in one go. This instruction can save plenty of computation time.

In H.264, there are 7 different block size types (from 16×16 to 4×4). We perform

SAD calculation for different block size types by using different methods as described below.

In 16×16 and 16×8 mode, there are 16 pixels in one line which are stored consecutively in memory. But WSAD instruction can only deal with 8 pixels at a time. So we have to load data to source register and destination register by two steps. Each step we load 64-bit data (8 pixels). Figure 4.5 shows the whole process. Only 7 instructions are needed to calculate the SAD of pixels in one line (4 load instructions, 2 WSAD instructions and 1 addition instruction).

In 8×16 and 8×8 and 8×4 modes, in each line there are 8 consecutive pixels. The data length satisfies the requirement of WSAD instruction very well. Only 3 instructions are used to calculate the SAD of pixels in one line (2 load instructions and 1 WSAD instruction). Figure 4.6 illuminates how to calculate SAD in this case.

For 4×8 and 4×4 modes, one line has 4 consecutive pixels (32 bits), so we can load two lines data to the register and combine them into a 64-bit data to satisfy the requirement of WSAD instructions. Figure 4.7 illuminates how to calculate SAD in this case.

In order to achieve the optimal optimization results, we design different SAD calculation functions for each mode. By using these assembly language modules, the complexity of motion estimation is reduced significantly.

4.5.2 Interpolation

H.264 supports 1/4 resolution motion estimation which can improve coding performance greatly. So the interpolation is one of the key components in H.264 encoder.

H.264 uses a 6-tap filter to interpolate the half pixel value. The coefficients of the filter are {1, -5, 20, 20, -5, 1}, the interpolation formula is shown in Eq. (4.7).

$$I_{half} = (A_{-2} - 5A_{-1} + 20A + 20A_1 - 5A_2 + A_3 + 16) >> 5 \tag{4.7}$$

where A_i are the neighboring integer pixels as shown in Figure 4.8. I_{half} is the interpolated half-pixel. $C_{1/4}$ and $D_{1/4}$ are the quarter-resolution pixels which can be calculated by the neighboring integer pixel or half pixel. For example, $C_{1/4}$ is the mean value of A and I_{half}, and $D_{1/4}$ is the mean value of I_{half} and A_1. From the above description, we can see that interpolation needs many multiplications, additions and

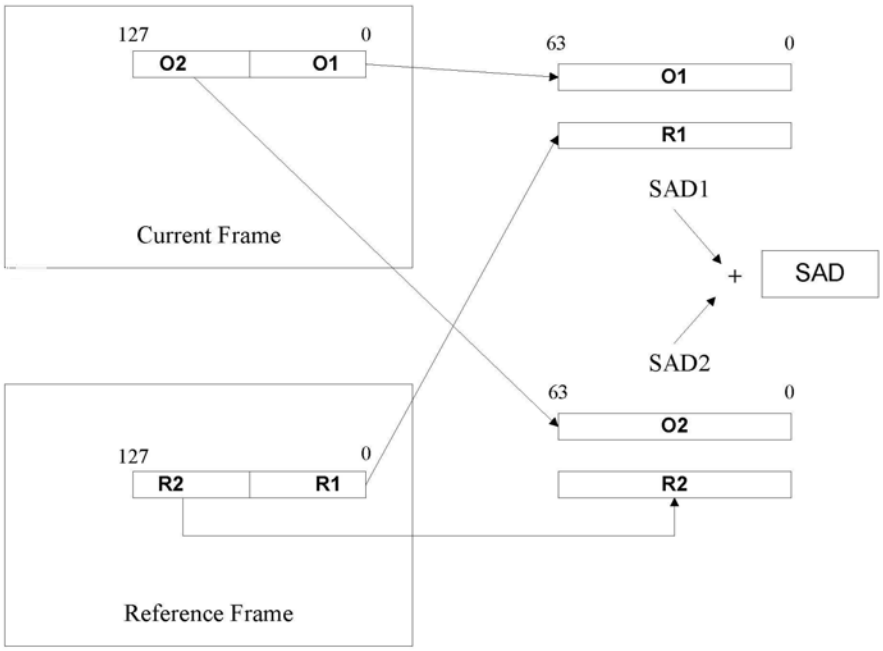

Figure 4.5: *Usage of WSAD instruction in* 16×16 *and* 16×8 *modes*

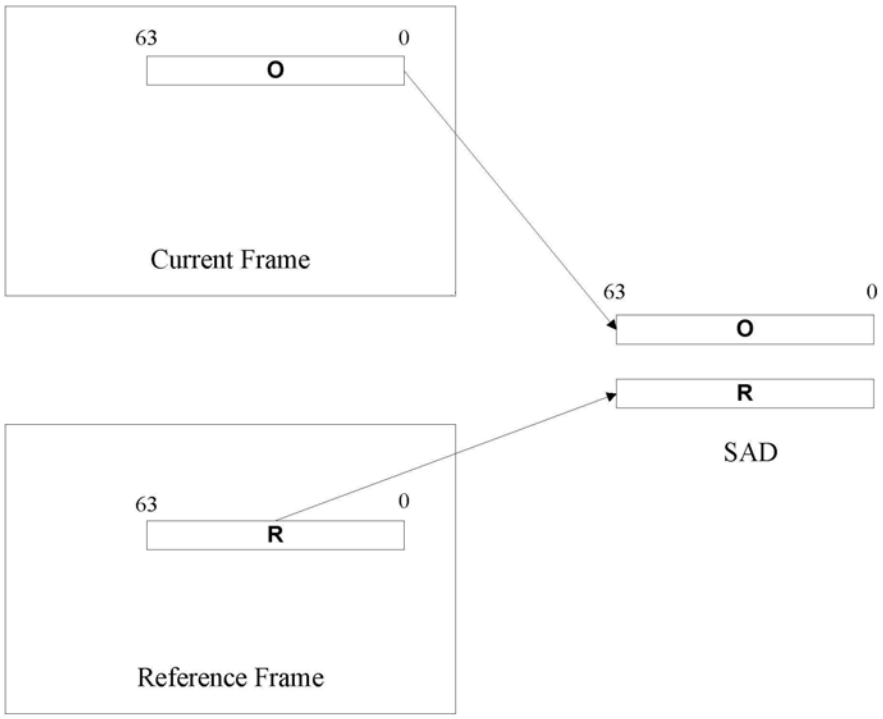

Figure 4.6: *Usage of WSAD instruction in* 8×16 *and* 8×8 *and* 8×4 *modes*

shift operations. So, the complexity of interpolation is very high and it is necessary to speed up this part.

In our implementation, we store 5I (I is the intensity value of each pixel) in a buffer

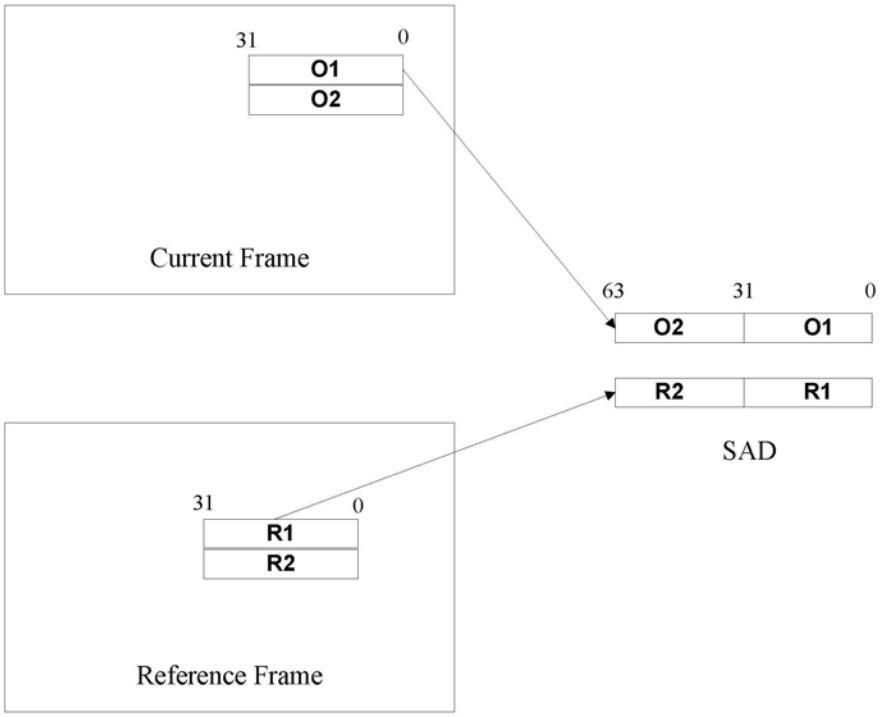

Figure 4.7: *Usage of WSAD instruction in* 4×8 *and* 4×4 *modes*

Figure 4.8: *Interpolation*

to reduce the repeated multiplication for the same pixel. The operation of multiplying 20 also can be replaced by shift operation to 5I. Figure 4.9 illuminates the whole process.

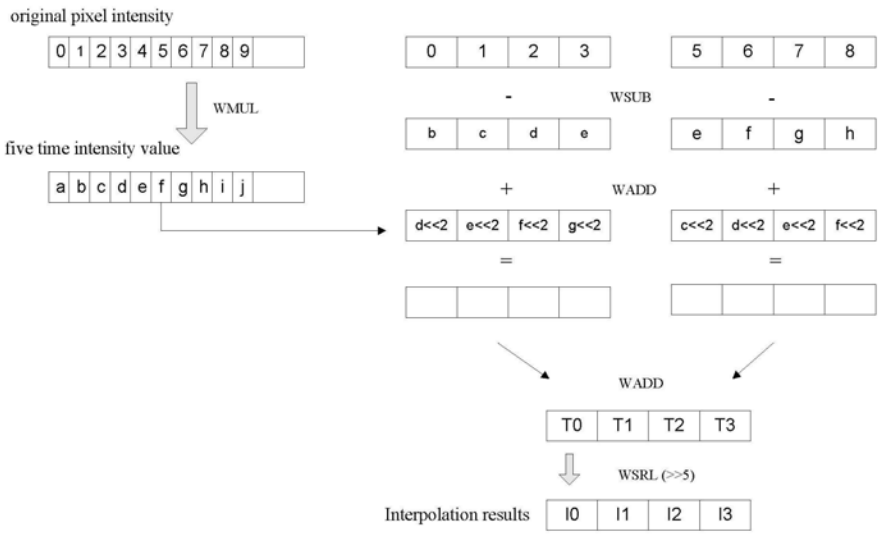

Figure 4.9: *Half pixel interpolation*

For the quarter pixel interpolation, because intensity value is the average value of two neighboring integer pixels or half pixels, we can use WAVG2 instruction (Two element averaged on unsigned vectors of 8-or 16-bit data) to carry out this operation conveniently.

4.5.3 Other Optimization Works

We also utilize the Wireless MMX instructions to rewrite some other time-consuming modules. The module involved is ICT (Integer Cosine Transform). Encoding time is reduced greatly after instruction optimization work.

4.6 Experimental Results

In this section, experimental results will be shown. In our experiments, six QCIF (176×144) test sequences are chosen, they are: Foreman, Carphone, Container, Grandma, MissA and Salesman. The performance of our optimized encoder and decoder will be shown in the following sub-sections.

4.6.1 Encoder

In order to evaluate the performance of our encoder, we compare our encoder with JM10.1 [HHI, 2005] and x264 [x264, 2004]. JM10.1 is the H.264 reference software released by ITU-T. This codec has the excellent coding performance but the code has many redundancies and the data structure is very complex. Therefore, the coding speed of JM is too slow to satisfy the requirement of real-time application. x264 is a well-known open source H.264 encoder. It is a very fast H.264 encoder and many commercial PC-based H.264 products use it as the core codec. The coding parameters are set as follows. QPs (quantization parameters) are set at 28, 32, 36 and 40. Hadamard transform and RDO are turned off. Frame rate is set as 30 frame per second. Coding performance is compared in terms of $\Delta PSNR$ and Δ*bit-rate* which are calculated by using the RD-curves fitting method [Bjontegaard, 2001].

All Intra Frames Mode

In this experiment, all 150 frames are coded as Intra frames. Table 4.1 tabulates the results. It can be seen that the average speed of our encoder is over 26 frames per

Sequence	QP=28	QP=32	QP=36	QP=40
Foreman	28.03	28.76	29.41	29.05
Carphone	26.28	26.35	27.9	27.84
Container	26.41	27.25	27.8	27.82
Grandma	25.79	26.92	27.15	27.19
MissA	26.52	26.6	26.8	26.84
Salesman	27.72	28.01	28.43	28.15
Average	26.79	27.32	27.92	27.82

Table 4.1: *Encoding speed in frame per second (all Intra frame, 150 frames)*

Sequence	Our encoder		x264	
	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]
Foreman	-0.92	0.06	1.26	-0.09
Carphone	-0.13	0.01	2.46	-0.17
Container	-3.03	0.21	-0.55	0.04
Grandma	-2.53	0.14	1.72	-0.09
MissA	-1.23	0.09	5.23	-0.34
Salesman	-1.21	0.08	1.17	-0.07
Average	-1.51	0.10	1.88	-0.12

Table 4.2: *Coding performance compared with JM10.1 (all Intra frame, 150 frames)*

second which can satisfy the requirement of real time applications.

Table 4.2 shows the coding performance comparison of JM10.1, x264 and our encoder. From the results, we observe that our encoder can achieve good coding efficiency and picture quality and is better than JM. Compared with x264, the performance of our encoder is much better.

Figure 4.10 plots the RD-curves of these three codecs.

IPPP... Mode

In this experiment, for total of 150 frames, the first frame is I-frame and all other following frames are coded as P-frames.The experimental results are shown in Table 4.3. From the results, we can see that our encoder is very fast, speed is from 50 fps to 75 fps under different QPs. The reason why the speed of inter coding is much faster than intra coding is because we use Pskip mode early detection technique in inter coding, so a lot of MBs with low motion are coded as Pskip mode. We also observe that with the increase of the quantization parameter, our encoder will be faster. That is because with large QPs, more DCT coefficients are quantized to zero and more MBs will be

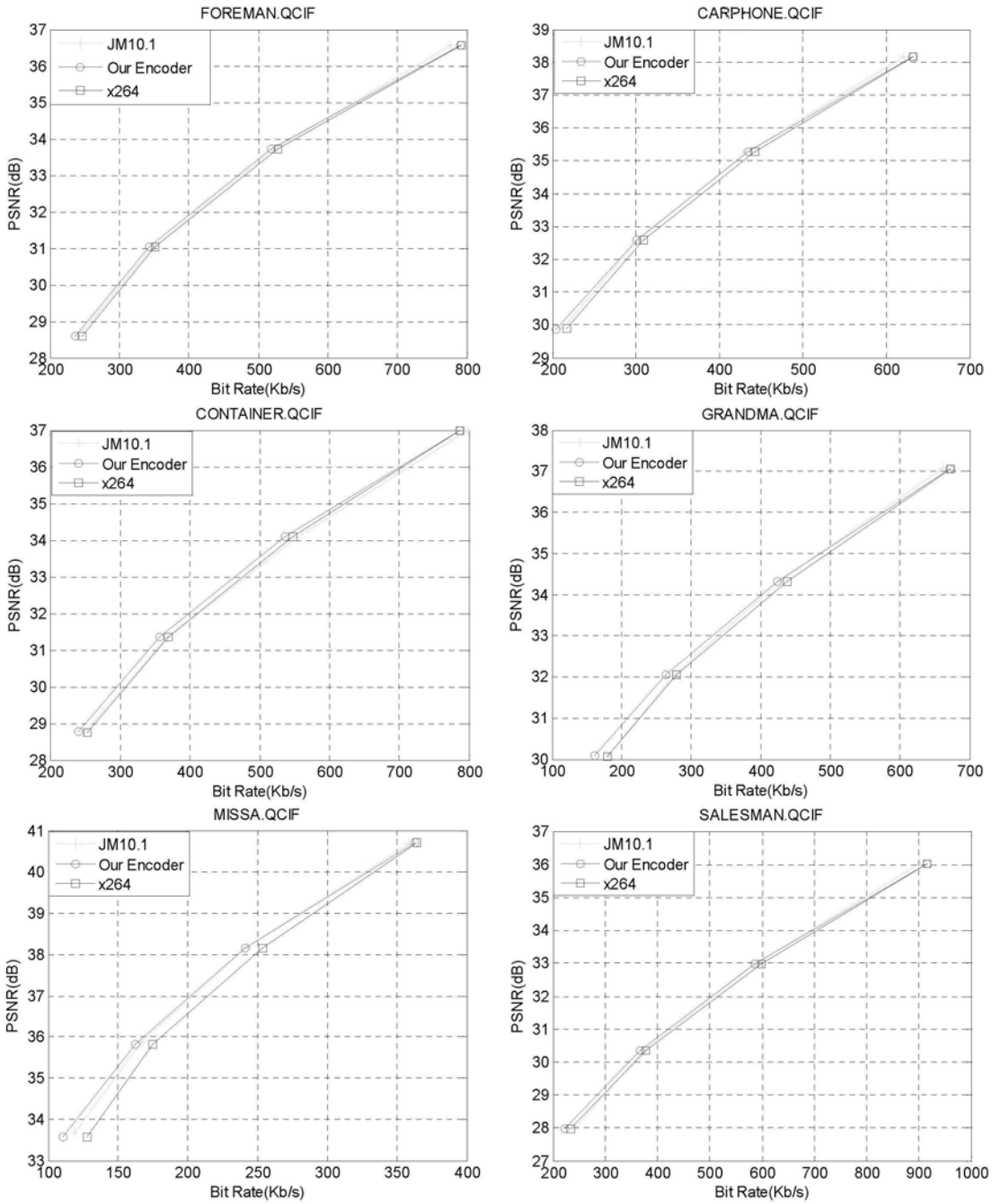

Figure 4.10: *Rate-distortion curves (all intra frames)*

decided to be Pskip mode.

Table 4.4 shows the coding performance. On average, the degradation of our encoder is only 0.22 dB in term of PSNR compared with JM10.1, or equivalently 5.94% increase in term of bit-rate. The performance of our encoder is better than that of x264. For the sequence MissA, the coding performance is even much better than JM10.1. That

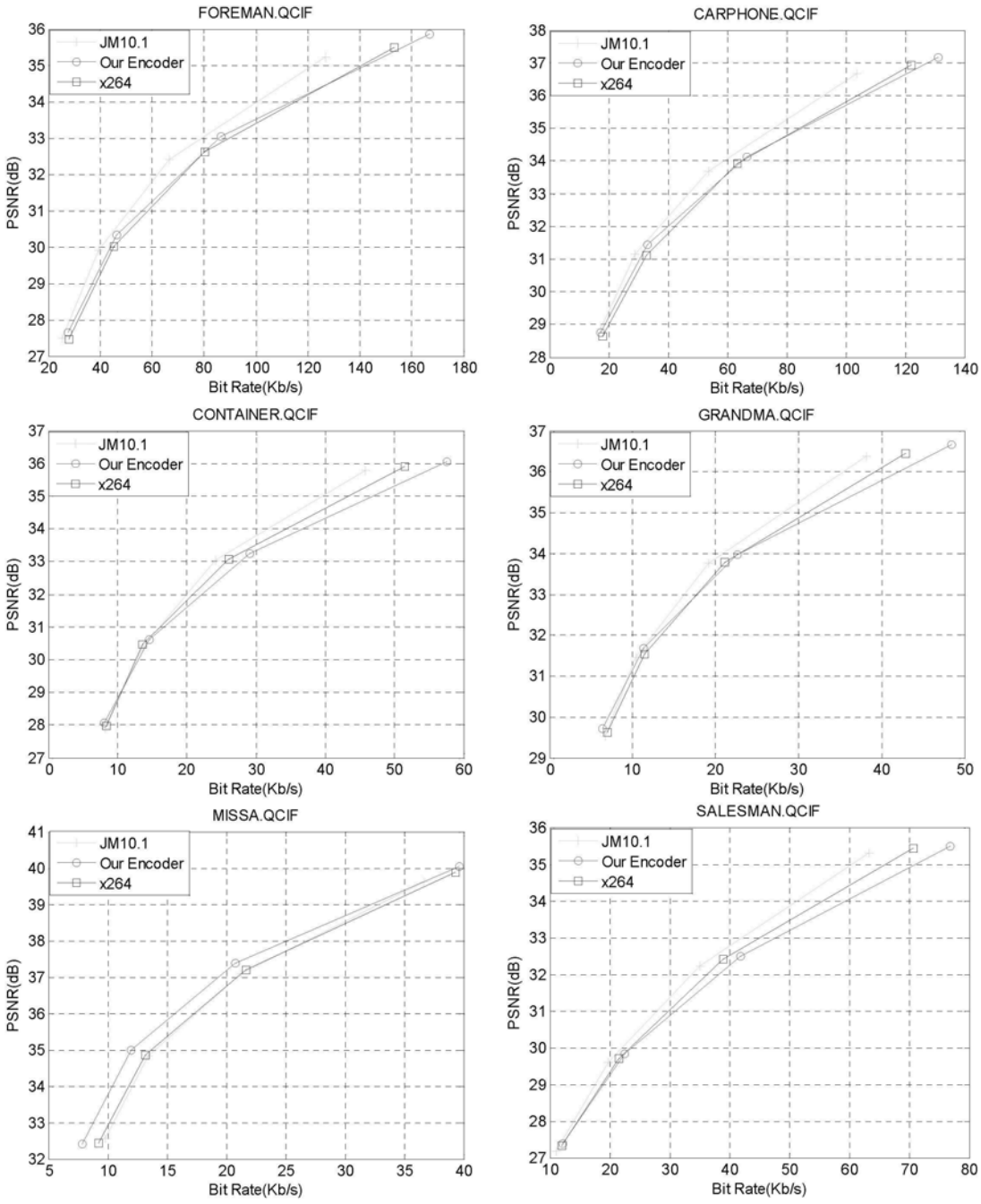

Figure 4.11: *Rate-distortion curves (IPPP... mode)*

is because plenty of MBs are coded as Pskip mode in this sequence and lots of bits are saved. Our encoder can maintain good coding quality in low bit-rates, this also matches the requirement of our application under mobile environment. Figure 4.11 plots the RD-curves of these three codecs.

Sequence	QP=28	QP=32	QP=36	QP=40
Foreman	28.67	34.09	40.58	49.34
Carphone	32.22	39.89	49.6	63.56
Container	62.29	73.53	82.6	89.29
Grandma	59.9	70.49	80.82	89.71
MissA	55.31	61.68	71.02	81.17
Salesman	56.82	63.56	72.67	83.33
Average	49.20	57.21	66.22	76.07

Table 4.3: *Encoding speed in frame per second (IPPP... mode, 150 frames)*

Sequence	Our encoder		x264	
	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]
Foreman	10.50	-0.47	13.54	-0.60
Carphone	9.52	-0.39	12.44	-0.51
Container	8.14	-0.35	3.53	-0.17
Grandma	7.18	-0.27	8.21	-0.30
MissA	-10.46	0.64	-1.34	0.07
Salesman	10.73	-0.47	6.79	-0.31
Average	5.94	-0.22	7.20	-0.30

Table 4.4: *Coding performance compared with JM10.1 (IPPPP...mode, 150 frames)*

4.6.2 Decoder

Table 4.5 shows the decoding speed under different QP values. Based on the results, it can be found that the speed of our optimized decoder is very fast when running on the mobile device. This decoder can satisfy the requirement of our application.

Sequence	QP=28 (F/s)	QP=32 (F/s)	QP=36 (F/s)	QP=40 (F/s)
Foreman	64.05	87.11	113.90	144.51
Carphone	74.89	103.45	141.91	185.87
Container	142.05	182.48	220.91	254.24
Grandma	144.23	186.10	227.96	266.90
MissA	143.68	179.86	212.77	242.72
Salesman	121.16	150.91	185.41	222.22
Average	115.01	148.32	183.81	219.41

Table 4.5: *Decoding speed (IPPPP...mode, 150 frames)*

4.7 H.264 Based Mobile Video Conferencing System

Based on our real time H.264 codec, an H.264 based mobile video conferencing system is finished.

Our system includes the following modules: video capture module, encoding module, wireless transmission module, decoding module, and the display module. The flow chart is shown in Figure 4.12. First, the video frame is captured by camera, and the captured frame is encoded by the encoder in PDA-I. The encoded bit-stream is then transmitted to PDA-II via the wireless channel. In PDA-II, the received bit-stream is decoded and the reconstructed frame is shown on the display. Our system works in the duplex mode.

Because the implementation is software based, our system can be easily transplanted on other mobile devices, such as mobile phone, laptop, etc. It also can be applied in other wired or wireless networks, such as 3G or 4G communication systems.

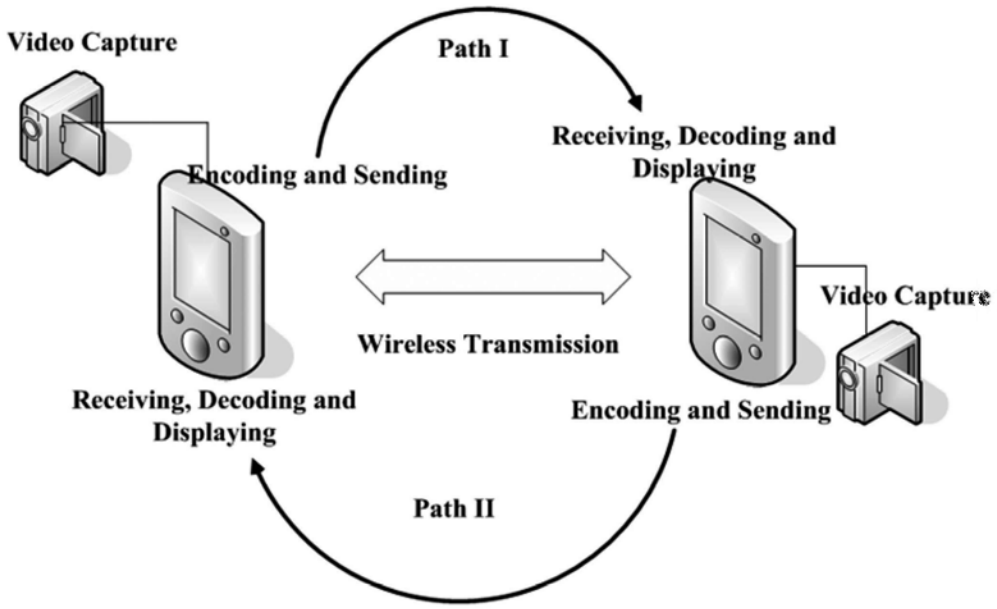

Figure 4.12: *Architecture of the H.264 based mobile video conferencing system*

4.8 Summary

This chapter introduces the work on the implementation of real-time H.264 Baseline Profile encoder and decoder on a mobile device. By performing program optimization,

algorithm optimization and instruction optimization, our codec can satisfy the requirements of real-time application in QCIF resolution and also achieves a good coding performance. Based on the real time H.264 codec, a wireless video conferencing system is designed. The key contributions of our work are the real-time implementation of the H.264/AVC encoder and decoder on mobile devices. It can be used in video conferencing, video telephony, video surveillance and other video-related applications.

All the work in this chapter was published in *IEEE Transactions on Consumer Electronics* as a full length article, entitled " Implementation of H.264 on Mobile Device".

Part II

PERCEPTUAL PICTURE CODING TECHNIQES

Chapter 5

Spatio-temporal Just Noticeable Distortion Model

5.1 Introduction

The last two decades have witnessed significant progress in image and video processing techniques, by which digital images are enhanced, compressed, transmitted, stored or verified before being displayed in front of the human eyes. Most of these techniques treat images and videos as 2-D or 3-D signals. In these methods, only statistical properties among the pixels are considered, but the perceptual features are often neglected. These mainstream signal-processing-based techniques could not explore the perceptual properties existing in the images very well. For example, the pixel-wise distortion metric, such as mean squared error (MSE) and peak signal-to-noise ratio (PSNR) are widely adopted in the traditional coding standards. But unfortunately, they have been widely criticized for not correlating well with perceived quality measurement [Girod, 1993]. Since the human visual system (HVS) is the ultimate receiver of the majority of processed images and videos, it is very important and advantageous to incorporate HVS into the image and video processing algorithms.

As we know, HVS cannot perceive all the changes in the images due to its underlying physiological and psychological mechanism, so a lot of perceptual redundancies exist in the processed images and videos. The removal of the perceptual redundancy has many advantages. Firstly, it ensures that only the visually important information is encoded or protected. Secondly, better compression performance can be achieved by discarding perceptually unnecessary information. How to model HVS more accurately and efficiently becomes an important and challenging research task. Just-noticeable distortion (JND) gives us a promising way to model the perceptual redundancy. JND refers to the maximum distortion which cannot be perceived by the human eyes. Knowledge on JND no doubt can guide the image/video processing algorithms and systems. JND can

be adopted in designing visual quality evaluation metric for images and videos [Sarnoff, 2003; Watson, 1993; Lin et al., 2003; Lu et al., 2005; Zhang et al., 2005; Ong et al., 2005], which are consistent with the HVS properties to achieve higher coding efficiency. In image and video compression, JND can be used to tune the quantizer and bit allocation [Chou and Li, 1995; Chou and Chen, 1996; Chin and Berger, 1999; Safranek and Johnston, 1989; Hontsch and Karam, 2002]. Thus, perceptually important information could be compressed with better quality and visual redundancy is removed efficiently. It was also reported in some literature that JND has been applied in motion estimation [Yang et al., 2005c], video conferencing [Yang et al., 2005a] and digital watermarking [Wolfgang et al., 1999; Zeng, 1999; Podilchuk and Zeng, 1998].

The computational JND model has been studied for a long time. Generally, the existing JND models belong to two categories. One is the model produced in image domain which is also called pixel-wise JND model. Another one is named subband JND model which is determined in transform domain, such as wavelet and DCT domains.

In [Chou and Li, 1995; Chou and Chen, 1996; Chin and Berger, 1999], pixel-based JND thresholds are determined according to the background luminance adaptation and the spatial contrast masking. Yang's JND profile [Yang et al., 2005b] is the improvement of [Chou and Chen, 1996], which deduces the overlapping effect between the luminance adaptation and the spatial contrast masking to achieve a more accurate JND map. In both [Chou and Chen, 1996] and [Yang et al., 2005b], the luminance difference between the successive frames is used to estimate the temporal JND threshold for videos. Although the pixel-wise JND model gives a more direct view of JND map of the original image/video, it does not incorporate the contrast sensitivity function (CSF) which describes the sensitivity of human vision for each frequency component, so this kind of model does not exploit the HVS completely.

Subband JND model is a hot research area since the CSF can easily be incorporated into the JND profile. [Safranek and Johnston, 1989] describes an early subband JND model, where JND threshold is determined for each subband by considering the luminance adaptation and the texture masking. In the famous Sarnoff JND model [Lubin, 1995], the JND map is produced by pyramid decomposition and some spatial filter banks. In [Watson et al., 1997], JND is generated in the wavelet domain. Because most of image and video compression schemes are performed in DCT domain, such as

H.261/3 and MPEG1/2/4, the DCT-based JND model attracts the interest of many researchers. A well-cited JND model in DCT domain was proposed by Ahumada and Perterson [Ahumada and Peterson, 1992], which gives the JND threshold for each DCT component by incorporating the spatial CSF. This scheme becomes the basis of many other JND models. The DCTune model [Watson, 1993] is an improvement of Ahumada's model, where the luminance adaptation and the contrast masking effect are added to the base threshold. In [Hontsch and Karam, 2002], DCTune model was modified to work with foveal region instead of a single pixel and was applied in perceptual image coding. More recently, Zhang [Zhang et al., 2005] proposed an improved model based on the DCTune model, where more realistic luminance adaptation was considered and block classification was combined with the contrast masking to achieve better performance. However, all above JND profiles are only effective for images and not for videos, because the temporal characteristics of the HVS are not considered.

In order to generate a complete JND profile for videos, not only the spatial CSF, but also the temporal CSF should be incorporated into the JND model. A computational spatio-temporal CSF model was proposed by Kelly [Kelly, 1979] based on his experimental data collected for the retinally stabilized travelling wave stimuli. Dally [Daly, 1998] improved this model by considering the retina movement compensation. Based on Kelly and Dally's models, Jia [Jia et al., 2006] estimated the JND thresholds for video by combining other visual effects, such as the luminance adaptation and contrast masking. In Jia's model, only the magnitude of motion contributes to the final spatio-temporal JND threshold, but the directionality of motion is neglected. However, as we know, motion is a vector. For two motion vectors which have the same magnitude but different directions, they will cause different temporal effects on a 2-D spatial frequency [Wang et al., 2002]. Therefore, it is not reasonable to ignore the directionality of motion in JND model.

With the development of the economy, High Definition (HD) TV will become more and more popular. The perceptual features hidden behind the larger images are a little different from the smaller images. However, all above discussed JND models are evaluated on smaller images and videos, so it is necessary to test the performance of JND profile on HD images or videos. It will benefit many applications, such as transparent HD video coding, etc.

In order to fix the problems discussed above, a spatio-temporal JND Profile is proposed in this chapter. This model estimates the explicit JND threshold in DCT domain. It not only incorporates the spatial and temporal CSF, but also considers all major vision effects, i.e., luminance adaption, contrast masking and the retina movement compensation. The main contributions of this work include:

- An psychophysical experiment is designed to determine the parameters of the spatial CSF, resulting in a more accurate JND model.
- Gamma correction is also considered to compensate for the Weber-Fechner law [Nettravali and Haskell, 1988] and more precise luminance adaptation funtion is obtained.
- A novel block classification method is proposed, which classifies the blocks into three types: edge, smooth and texture block. For each block type, different contrast masking effect is applied.
- A new temporal modulation factor is introduced, which incorporates the temporal CSF and retina movement compensation. Moreover, the directionality of motion is also considered in this model.
- The proposed JND profile is evaluated on the HD images and videos to confirm the reliability of this model.

Experimental results show that the proposed model is consistent with human visual system. Compared with the other JND profiles, the proposed model can tolerate more distortion and have much better perceptual quality. Since this model is generated in the DCT domain, it is easily applied to many image and video processing algorithms, such as JPEG, MPEG1/2/4, H.261/263/264 and AVS, etc.

The rest of the chapter is organized as follows. Some related JND models are introduced in Section 5.2. In Section 5.3, the main structure of the proposed JND profile is introduced. The spatial JND model is also presented, which includes the CSF effect, parameterization of the model, luminance adaptation effect and contrast masking effect. Temporal JND model is described in Section 5.4. The experimental results are shown and discussed in Section 5.5. Section 5.6 contains the concluding remarks.

5.2 Related JND Models

In this section, we present a brief overview of previous research work on JND models, including the pixel-wise JND models and the subband JND models.

5.2.1 Pixel-wise JND Models

Pixel-wise JND estimation models [Chou and Li, 1995; Chou and Chen, 1996; Chin and Berger, 1999; Yang et al., 2005b; Ramasubramanian et al., 1999] exploit the JND threshold in pixel domain, which have been applied for motion estimation [Yang et al., 2005c], quality evaluation [Lin et al., 2003] and other image/video processing field [Chin and Berger, 1999; Ramasubramanian et al., 1999; Yang et al., 2005a]. Two famous spatio-temporal pixel-based JND models proposed by Chou et al. [Chou and Li, 1995; Chou and Chen, 1996] and Yang et al. [Yang et al., 2005b] are briefly introduced in this section.

Chou's Model

In Chou's model, the spatial JND threshold is related to two components, the luminance masking factor and the contrast masking factor. It is calculated by the following equation:

$$JND_s(x,y) = max\{f_1(mg(x,y)),\ f_2(bg(x,y))\}, \quad for\ 0 \le x \le H,\ 0 \le y \le W \quad (5.1)$$

where JND_s stands for the spatial JND threshold. f_1 represents the contrast masking effect function and f_2 is utilized to compute the visibility threshold due to the average background luminance masking. H and W denote the image height and width, respectively. $mg(x,y)$ and $bg(x,y)$ represent the weighted average of luminance changes and the average background luminance around the pixel (x,y), respectively.

The function f_1 is approximated by Eq. (5.2),

$$f_1(mg(x,y)) = mg(x,y) \times \beta, \quad for\ 0 \le x \le H,\ 0 \le y \le W \quad (5.2)$$

where the parameter β is the slope of the linear function and $mg(x,y)$ across the pixel (x,y) is determined by:

$$mg(x,y) = max\{|grad_k(x,y)|\} \quad k = 1,2,3,4 \tag{5.3}$$

$$grad_k(x,y) = \frac{1}{16}\sum_{i=1}^{5}\sum_{j=1}^{5} p(x-3+i, y-3+j)\cdot G_k(i,j) \tag{5.4}$$

where $p(x,y)$ denotes the pixel value at (x,y). The four mask operators, $G_k(i,j)$, for $k = 1, \cdots 4$, are shown in Figure 5.1.

0	0	0	0	0
1	3	8	3	1
0	0	0	0	0
-1	-3	-8	-3	-1
0	0	0	0	0

G_1

0	0	1	0	0
0	8	3	0	0
1	3	0	-3	-1
0	0	-3	-8	0
0	0	-1	0	0

G_2

0	0	1	0	0
0	0	3	8	0
-1	-3	0	3	1
0	-8	-3	0	0
0	0	-1	0	0

G_3

0	1	0	-1	0
0	3	0	-3	0
0	8	0	-8	0
0	3	0	-3	0
0	1	0	-1	0

G_4

Figure 5.1: *Operators for calculating the weighted average of luminance changes in four directions [Chou and Li, 1995]*

The background luminance masking factor f_2 is calculated based on a U-shape function, which means that JND threshold takes a lower value at mid-range grey level and becomes higher in the dark or bright area. The U-shape function was given by Chou and Li [Chou and Li, 1995] to describe the luminance adaptation effect.

$$f_2(x,y) = \begin{cases} 17\cdot(1-(\frac{bg(x,y)}{127})^{1/2})+3 & bg(x,y) \le 127 \\ \frac{3}{128}\cdot(bg(x,y)-127)+3 & otherwise \end{cases} \tag{5.5}$$

where $bg(x,y)$ is the average background luminance around the pixel (x,y) and is calculated by a weighted low-pass operator. This low-pass filter is modeled by a 5×5 matrix.

In [Chou and Chen, 1996], Chou et al. extended the JND model to temporal domain by introducing a temporal modulation function. The spatio-temporal pixel-wise JND threshold JND_{S-T} is calculated by:

$$JND_{S-T}(x,y,n) = f_3(ild(x,y,n))\cdot JND_s(x,y,n) \tag{5.6}$$

where $ild(x,y,n)$ stands for the average interframe luminance difference between the

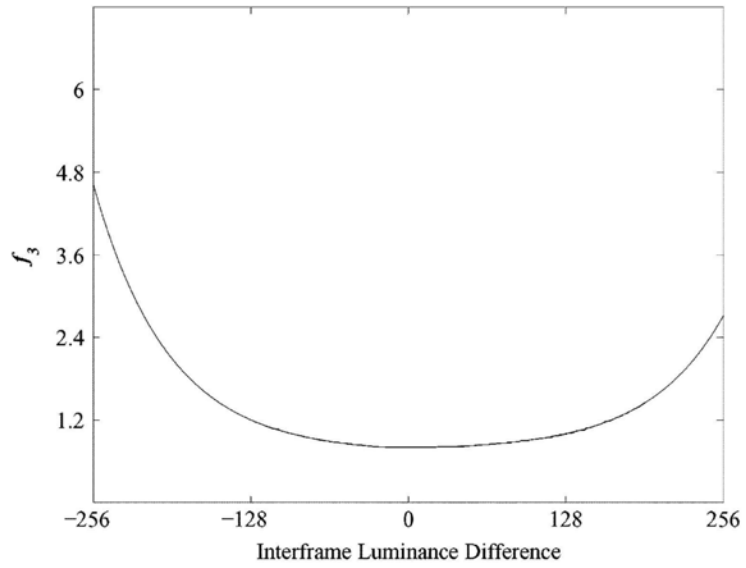

Figure 5.2: *Temporal masking effect in Chou's JND model [Chou and Chen, 1996]*

n_{th} and the $(n-1)_{th}$ frame. It is calculated by:

$$ild(x,y,n) = (p(x,y,n) - p(x,y,n-1) + bg(x,y,n) - bg(x,y,n-1))/2 \tag{5.7}$$

The empirical curve of the temporal modulation function f_3 is shown in Figure 5.2. For the stationary area, the scale factor is only 0.8 as $|ild(x,y,n)| \leq 5$. For the moving area, since human eyes are not sensitive for the distortion, the scale factor will take larger value to make the JND threshold higher.

Yang's Model

Yang et al. improved Chou's model in [Yang et al., 2005b]. In Yang's model, a non-linear additivity model for masking (NAMM) is introduced to reduce the overlapping part between the luminance and texture masking. Moreover, the texture masking is distinguished in the edge regions and the non-edge regions since the distortion around the edge is easier to be noticed than that in other areas. Thus, Yang's model can estimate JND threshold more accurately and properly. The spatial JND of each pixel can be calculated by the following equation:

$$JND_s(x,y) = T_l(x,y) + T_t(x,y) - C_{l,t} \cdot min\{T_l(x,y), T_t(x,y)\} \tag{5.8}$$

where $T_l(x,y)$ and $T_t(x,y)$ are the visibility threshold for background luminance masking and texture masking, respectively; and $C_{l,t}$ accounts for the overlapping effects between two masking effects and takes the value ranging from 0 to 1.

For the background luminance masking function $T_l(x,y)$, it is same as the f_2 in Chou's model (as shown in Eq. (5.5)). The texture masking function $T_t(x,y)$ considers the difference between edge and non-edge area and is obtained by Eq. (5.9),

$$T_t(x,y) = \eta \cdot G(x,y) \cdot W_e(x,y) \tag{5.9}$$

where η is a control parameter; $G(x,y)$ is just the $mg(x,y)$ in Chou's model (as shown in Eq. (5.3)). $W_e(x,y)$ is an edge-related weight of the pixel at (x,y), and the matrix W_e is computed by edge detection followed by a Gaussian low-pass filter

$$W_e(x,y) = L * h \tag{5.10}$$

where L is the edge map of the image generated by Canny edge detector [Canny, 1986]. h is a $k \times k$ Gaussian low-pass filter with the standard deviation σ.

Yang's model borrows the temporal masking modulation function in Chou's model to extend the spatial JND model to the temporal domain. The spatio-temporal JND threshold JND_{S-T} is also obtained by Eq. (5.6).

5.2.2 Subband JND Models

Subband JND models [Safranek and Johnston, 1989; Lubin, 1995; Ahumada and Peterson, 1992; Peterson et al., 1993a; Watson, 1993; Watson et al., 1997; Hahn and Mathews, 1998; Tong and Venetsanopoulos, 1998; Daly, 1998; Hontsch and Karam, 2000; Hontsch and Karam, 2002; Zhang et al., 2005; Jia et al., 2006] are more consistent with HVS since the CSF can easily be incorporated into the JND profile.

More intensive research in subband JND has been exploited in DCT domain [Ahumada and Peterson, 1992; Peterson et al., 1993a; Watson, 1993; Hontsch and Karam, 2000; Hontsch and Karam, 2002; Zhang et al., 2005; Jia et al., 2006], because most of image and video compression schemes are performed in DCT domain, such as H.261/3 and MPEG1/2/4. The proposed JND model which will be introduced in this chapter is

also generated in DCT domain. In this section, three well-cited DCT-based JND models will be briefly introduced, which are Ahumada & Peterson's CSF model [Ahumada and Peterson, 1992], DCTune model [Watson, 1993] and Zhang's model [Zhang et al., 2005].

Ahumada & Peterson's Model

A well-known JND model in DCT domain was developed by Ahumada and Peterson [Ahumada and Peterson, 1992], based on a spatial CSF that describes the sensitivity of HVS versus spatial frequencies. This scheme becomes the basis of many other JND models [Watson, 1993; Hontsch and Karam, 2002; Zhang et al., 2005]. This model considers some visual factors, such as spatial frequency, orientation, mean display luminance and other parameters. The formulae for calculating the JND threshold of digital image $T_{DCT}(n,i,j)$ for each DCT coefficient in the n_{th} block are listed below.

$$T_{DCT}(n,i,j) = \frac{M \cdot T(n,i,j)}{\phi_i \phi_j (L_{max} - L_{min})} \tag{5.11}$$

where T_{DCT} and T stand for the JND value in grey level and in luminance, respectively. n is the index of a block, and i and j are the DCT coefficients' indices ($i,j = 0$ to 7). $M = 256$ is the number of grey levels for an 8-bit image, L_{max} and L_{min} are the maximum and the minimum display luminance, ϕ_i and ϕ_j are the DCT coefficient normalizing factors and can be determined by

$$\phi_m = \begin{cases} \sqrt{1/N}, & m = 0 \\ \sqrt{2/N}, & m > 0 \end{cases} \tag{5.12}$$

The function of Eq. (5.11) is to convert the luminance values to corresponding gray levels since the JND threshold refers to the change of the intensity of the input digital image, not the luminance value. The JND value in luminance $T(i,j)$ is calculated by a parabola equation:

$$logT(i,j) = log(\frac{T_{min}}{r + (1-r)cos^2\theta_{i,j}}) + K(logf_{i,j} - logf_{min})^2 \tag{5.13}$$

where r is empirically set to 0.6. $f_{i,j}$ is the spatial frequency of the DCT coefficient

with the location (i, j) and can be determined by:

$$f_{i,j} = \frac{1}{2N}\sqrt{(i/\omega_x)^2 + (j/\omega_y)^2} \tag{5.14}$$

where $N = 8$. ω_x and ω_x stand for the horizontal and vertical visual angle of a pixel, which is based on the viewing distance D and the width/length of a pixel on the monitor Λ. If we assume the horizontal and vertical size of a pixel are the same, it can be calculated from Eq. (5.15).

$$\omega_\hbar = 2 \cdot arctan(\frac{\Lambda}{2 \cdot D}) \qquad (\hbar = x, y) \tag{5.15}$$

$\theta_{i,j}$ stands for the directional angle of the corresponding DCT coefficient and can be calculated by:

$$\theta_{i,j} = arcsin\frac{2f_{i,0}f_{0,j}}{f_{i,j}^2} \tag{5.16}$$

T_{min}, f_{min} and K are determined by:

$$T_{min} == \begin{cases} L^{a_T} L_T^{1-a_T}/S_0, & L \leq L_T \\ L/S_0, & L > L_T \end{cases} \tag{5.17}$$

where $L_T = 13.45cd/m^2$, $S_0 = 94.7$ and $a_T = 0.649$.

$$f_{min} == \begin{cases} f_0 L^{a_f} L_f^{-a_f}, & L \leq L_f \\ f_0, & L > L_f \end{cases} \tag{5.18}$$

where $f_0 = 6.78cycle/degree$, $a_f = 0.182$ and $L_f = 300cd/m^2$.

$$K == \begin{cases} K_0 L^{a_K} L_K^{-a_K}, & L \leq L_K \\ K_0, & L > L_K \end{cases} \tag{5.19}$$

where $K_0 = 3.125$, $a_K = 0.0706$ and $L_K = 300cd/m^2$.

Ahumada & Peterson's model is an image-independent perceptual (IIP) model because the JND threshold is computed independent of any images. The fundamental drawback of this model is that it ignores the local features of each image, so the JND matrix is fixed for all the blocks of every image.

DCTune Model

DCTune model is a well-developed DCT-based JND model derived by Watson [Watson, 1993] from the NASA Vision Group. This model is an improvement of Ahumada & Peterson's model by incorporating two visual effects, which are the luminance masking and the contrast masking. It is an image-dependent perceptual (IDP) JND model since the local characteristics exiting in each image are considered.

The luminance masking threshold models the dependence of the detection threshold and local image mean luminance. The brighter the background, the higher the luminance threshold. In Ahumada & Peterson's model, the contrast sensitivity is approximated by the parabolic equations and affected by the background luminance. However, more accurate contrast sensitivity should be estimated block by block since the local mean luminance of each block is different. Watson gave a simpler solution to approximate the dependence of JND threshold upon the local mean luminance with a power function:

$$T_{i,j,k} = T_{i,j}(c_{0,0,k}/\bar{c}_{0,0})^{\alpha_T} \tag{5.20}$$

where $T_{i,j,k}$ is the modified JND threshold by luminance masking. $T_{i,j}$ is the JND threshold calculated by Ahumada & Peterson's model and is the $T_{DCT}(n,i,j)$ in Eq. (5.11). $\bar{c}_{0,0}$ is a constant corresponding to the display luminance and is set to be 1024 for an 8-bit image. $c_{0,0,k}$ is the DC value in the 8×8 block. α_T is suggested to be 0.649.

Contrast masking is an important phenomenon in the HVS perception and is referred to as the reduction in the visibility of one visual component in the presence of another. Watson derived a equation for calculating the contrast masking based on Legge and Foley's model [Legge and Foley, 1980]. The final masked JND threshold modulated by contrast masking effect is calculated by:

$$M_{i,j,k} = max[T_{i,j,k}, |c_{i,j,k}|^{\epsilon}(T_{i,j,k})^{1-\epsilon}] \tag{5.21}$$

where ϵ takes a value of 0.7 empirically.

Zhang's Model

Zhang et al. [Zhang et al., 2005] proposed a method to estimate the JND threshold based on the HVS for the block based DCT. The model incorporates the spatial CSF, improved luminance adaptation effect, intra-frequency and inter-frequency masking effect, which is formulated as

$$T_{JND}(n,i,j) = T_{Base}(n,i,j) \times a_{Lum}(n) \times a_e(n,i,j) \tag{5.22}$$

where n is the index of a block, and i and j are the DCT coefficients' indices ($i, j = 0$ to 7). T_{Base} represents the base threshold which is obtained from Ahumada & Peterson's model and is the T_{DCT} in Eq. (5.11). a_{Lum} and a_e stand for the luminance adaptation factor and the contrast masking factor, respectively.

For luminance adaptation, a constant value of 128 could be used as background luminance value for 8-bit images. However, this tends to underestimate the visibility threshold in dark and bright regions of an image. Thus, a modulation factor was proposed to overcome the limitation:

$$a_{Lum}(n) == \begin{cases} k_1(1 - 2C(n,0,0)/(GN))^{\lambda_1} & C(n,0,0) \leq (GN)/2 \\ k_2(2C(n,0,0)/(GN) - 1)^{\lambda_2} & C(n,0,0) > (GN)/2 \end{cases} \tag{5.23}$$

where $k_1 = 2$, $k_2 = 0.8$, $\lambda_1 = 3$ and $\lambda_2 = 2$. G is the total number of grey levels ($G = 256$ for 8-bit images). N stands for the dimension of the block and takes a value of 8 here. $C(n,0,0)$ is the DC component of the n_{th} block and represents the average local intensity.

The contrast masking depends on the local features of the image. The human eyes are sensitive to the distortion in smooth regions, but not sensitive in the texture regions. The sensitivity for edge region lies in between. Therefore, contrast masking should be discriminated for different image contents.

In Zhang's model, a DCT block can be divided into four parts: DC, low-frequency (LF), medium-frequency (MF), and high-frequency (HF) parts, as shown in Figure 5.3. L, M and H denote the sums of the absolute DCT coefficients in LF, MF and HF,

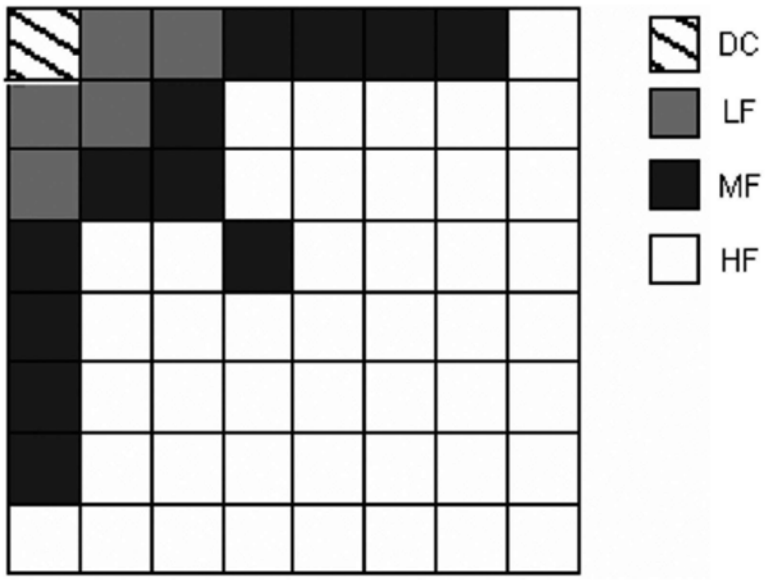

Figure 5.3: *Block classification scheme for a DCT block [Zhang et al., 2005]*

respectively. The block energy $TexE$ is calculated by:

$$TexE = M + H \tag{5.24}$$

Blocks are classified as *Texture*, *Edge* and *Plain* blocks according to Tong's block classification method [Tong and Venetsanopoulos, 1998]. Then contrast masking factor a_e can be calculated as:

$$a_e(n,i,j) = a_{inter}(n) \times a_{intra}(n,i,j) \tag{5.25}$$

$$a_{inter}(n) = \begin{cases} 1 + [(TexE(n) - \xi_1)/(2\xi_2 - \xi_1)]\delta_1, & \text{for Texture block} \\ \delta_1, & \text{for Edge block and } L + M > 400 \\ \delta_2, & \text{for Edge block and } L + M \leq 400 \\ 1, & \text{for Plain block} \end{cases} \tag{5.26}$$

where $\xi_1 = 290$, $\xi_2 = 900$, $\delta_1 = 1.25$ and $\delta_2 = 1.125$.

$$a_{intra}(n,i,j) = \begin{cases} 1, & \text{for}(i,j) \in LF \cup MF \text{in Smooth and Edge bloc} \\ max\{1, [\frac{C(n,i,j)}{T_{Base}(n,i,j) \times a_{Lum}(n,i,j)}]^{\epsilon}\}, & otherwise \end{cases} \tag{5.27}$$

where $C(n,i,j)$ is the DCT coefficient, and $\epsilon = 0.36$.

5.3 Proposed JND Model

JND can be considered as the response of the HVS which filters the input images. Since the convolution in spatial domain is equivalent to the multiplication in transform domain, the JND in the DCT domain is typically expressed as a product of a base threshold and some modulation factors [Watson, 1993; Hontsch and Karam, 2002; Zhang et al., 2005; Jia et al., 2006]. k is the index of a frame in the video sequences and n is the index of a block in the k_{th} frame, and i and j are the DCT coefficients' indices ($i, j = 0$ to 7). Then the corresponding JND can be expressed as:

$$T_{JND}(k,n,i,j) = T_{JND_s}(k,n,i,j) \times F_T(k,n,i,j) \tag{5.28}$$

$$T_{JND_s}(k,n,i,j) = T_{Basic}(k,n,i,j) \times F_M(k,n,i,j) \tag{5.29}$$

$$F_M(k,n,i,j) = F_{lum}(k,n) \times F_{contrast}(k,n,i,j) \tag{5.30}$$

where $T_{JND}(k,n,i,j)$, $T_{JND_s}(k,n,i,j)$ and $F_T(k,n,i,j)$ are the spatio-temproal JND threshold, the spatial JND and the temporal modulation factor, respectively. $T_{Basic}(k,n,i,j)$ is the base threshold which is generated by the spatial contrast sensitivity function (CSF). The modulation factor $F_M(k,n,i,j)$ is the product of the luminance adaptation factor F_{lum} and the contrast masking factor $F_{contrast}$.

In the following part of this section, the spatial JND profile (T_{JND_s}) in Eq. (5.29) is proposed first. It is based on the spatial contrast sensitivity function (CSF), the luminance adaptation effect and the contrast masking effect.

5.3.1 Spatial CSF Effect

Human eyes show a band-pass property in the spatial frequency domain. Various CSF models [Daly, 1992; Kelly, 1979; Foley and Boynton, 1994; Daly, 1998; Ngan et al.,

1989; van den Branden Lambrecht and Kunt, 1998; Nill, 1985] have been introduced in the past decades. Ngan et al. [Ngan et al., 1989] and Nill [Nill, 1985] showed that the generalized HVS models (as shown in Figure 5.4) can be expressed by

$$H(\omega) = (a + b\omega)exp(-c\omega) \tag{5.31}$$

where a, b and c are constants.

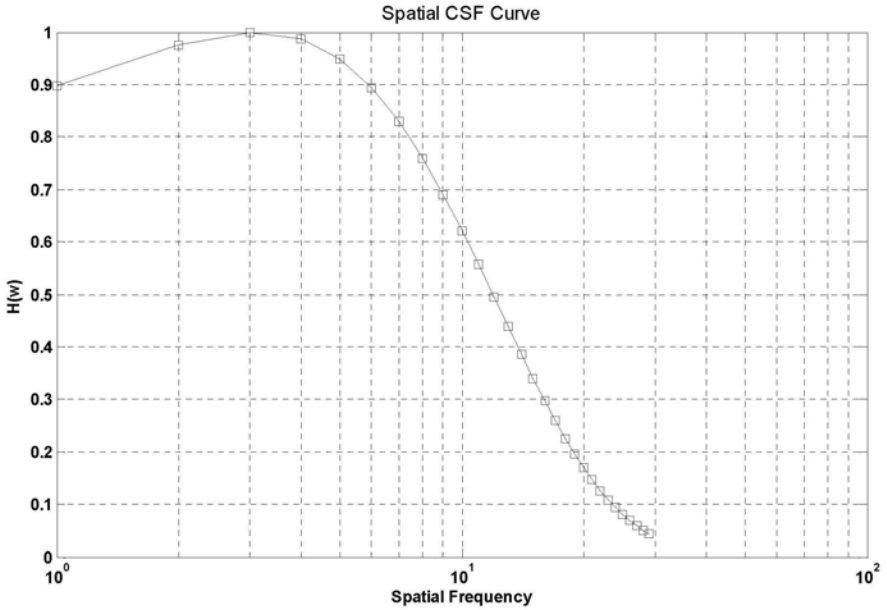

Figure 5.4: *Spatial CSF curve*

In [Wang et al., 2002], the definition of spatial contrast sensitivity is described as follows:

$$\psi(x, y, t) = B(1 + mcos(2\pi f_x x)) \tag{5.32}$$

where $\psi(x, y, t)$ is a spatial stimulus signal. For a fixed mean brightness level B and frequency f_x, the modulation level m was varied and the viewer was asked to identify the lowest modulation level m_{min} at which the spatial change became just noticeable. $1/m_{min}$ is defined as spatial contrast sensitivity.

Because the sensitivity modeled by Eq. (5.31) is the inverse of the JND threshold based on the definition of the contrast sensitivity in Eq. (5.32), the base JND threshold for a specified spatial frequency ω can be simply expressed by Eq. (5.33):

$$T(\omega) = exp(c\omega)/(a + b\omega) \tag{5.33}$$

where ω is the spacial frequency (cycle/degree). For the $(i,j)_{th}$ subband in the DCT block, the corresponding frequency ω_{ij} can be calculated by Eq. (5.34):

$$\omega_{ij} = \frac{1}{2N}\sqrt{(i/\theta_x)^2 + (j/\theta_y)^2} \tag{5.34}$$

$$\theta_\hbar = 2 \cdot arctan(\frac{\Lambda_\hbar}{2 \cdot l}) \qquad (\hbar = x, y) \tag{5.35}$$

where N is the dimension of the DCT block (is 8 in this case), θ_x and θ_y are the horizontal and vertical visual angles of a pixel and they can be calculated by Eq. (5.35). l is the viewing distance and Λ stands for the display width/length of a pixel on the monitor. According to the international standard ITU-R BT.500-11 [ITU-R, 2002] (Methodology for the subjective assessment of the quality of television pictures), the ratio of viewing distance to picture height should be a fixed number which is usually from 3 to 6 depending on the picture size. Moreover, for most of the displays, PAR (pixel aspect ratio) is equal to 1. This means that the horizontal and vertical visual angles (θ_x, θ_y) are identical. Then Eq. (5.35) leads to Eq. (5.36) easily. Here, R_{vd} stands for the ratio of viewing distance to picture height. Pic_h is the number of pixels in picture height. From this equation, it can be found easily that as the visual angle decreases with increasing picture size, each DCT component will represent higher spatial frequency, resulting in higher JND threshold.

$$\theta_x = \theta_y = 2 \cdot arctan(\frac{1}{2 \times R_{vd} \times Pic_h}) \tag{5.36}$$

When Eq. (5.33) is used for predicting the distortion threshold, several factors need to be considered:

1. Human visual sensitivity has directionality [Peterson et al., 1993a; Ahumada and Peterson, 1992]. Usually, the eye is sensitive to the horizontal and vertical frequency components (i or $j = 0$) and not so sensitive to the diagonal components. This is called *oblique* effect.

2. Visual system summation of the distortion over a spatial frequency range also

needs to be considered. Such spatial summation causes a decrease in threshold values [Peterson et al., 1993a; Ahumada and Peterson, 1992]

With all the considerations mentioned above, the *oblique* effect factor and the *spatial summation* effect factor were first introduced in Ahumada's JND model [Peterson et al., 1993a; Ahumada and Peterson, 1992]. These two factors are also used in the proposed JND model, thus the base threshold for a DCT subband in Eq. (5.33) is modified as:

$$T'(n,i,j) = \frac{1}{\phi_i \phi_j} \cdot \frac{exp(c\omega_{ij})/(a+b\omega_{ij})}{r+(1-r)\cdot cos^2\varphi_{ij}} \tag{5.37}$$

$$T_{Basic}(n,i,j) = s \cdot T'(n,i,j) \tag{5.38}$$

where parameter s is to account for the *spatial summation* effect and takes the value of 0.25 empirically [Peterson et al., 1993a]. ϕ_i and ϕ_j are DCT normalization factors.

$$\phi_m = \begin{cases} \sqrt{1/N}, & m = 0 \\ \sqrt{2/N}, & m > 0 \end{cases} \tag{5.39}$$

The term $1/(r+(1-r)\cdot cos^2\varphi_{ij})$ accounts for the *oblique* effect, where r is empirically set to 0.6 [Ahumada and Peterson, 1992] and φ_{ij} stands for the directional angle of the corresponding DCT component.

$$\varphi_{ij} = arcsin(\frac{2\omega_{i,0}\omega_{0,j}}{\omega_{ij}^2}) \tag{5.40}$$

It could be observed that when i and j are equal, the DCT coefficients are located in the oblique direction, which causes a further increase in JND threshold.

5.3.2 Parameterization of the Model

The model, as described in Eq. (5.37) needs to be parameterized. There are three parameters a, b and c to be determined. A perceptual experiment was performed to test the JND thresholds to some selected DCT basis functions. Here, $\hat{T_{\omega_{ij}}}$ is used to denote the JND threshold detected by the perceptual experiment for the spatial frequency ω_{ij}.

This psychophysical experiment is described as follows: For a 720 × 720 image, whose pixel intensities are all 128, distortion was added to some selected DCT frequency

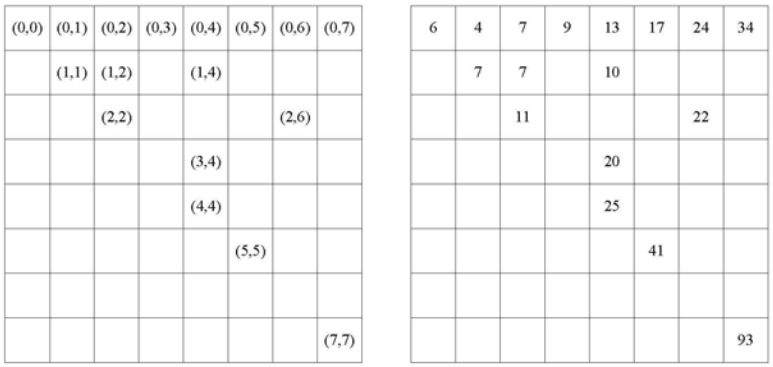

Figure 5.5*: Selected DCT frequency components and the detected thresholds.*

components individually. The selected DCT basis functions are shown in Figure 5.5 (a). For each tested DCT basis function, five amplitudes of distortion were chosen based on preliminary measurements of the sensitivity of the authors. A group of 15 viewers voted on whether the distortion was visible. When more than 50% of the viewers voted "visible", this distortion would be considered over the JND threshold. Then the JND threshold for each selected DCT basis function can be obtained as shown in Figure 5.5 (b).

Then the least mean squared error method is used to get the best fitted values of $(a,\ b,\ c)$ as :

$$(a, b, c) = argmin \sum_{\omega_{ij}} [\hat{T_{\omega_{ij}}} - T'(\omega_{ij}, a, b, c)]^2 \tag{5.41}$$

where $T'(\omega_{ij}, a, b, c)$ is calculated by Eq. (5.37). As the result, a=1.33, b=0.11, c=0.18.

In this experiment, the test equipment is a 22-inch Viewsonic professional series P225fb CRT display (contrast ratio is 450:1, maximum resolution is 2560×1920, maximum brightness is $110cd/m^2$).

5.3.3 Luminance Adaptation Effect

According to the Weber-Fechner law [Nettravali and Haskell, 1988], the minimally perceptible brightness difference increases with the increase of the background brightness. This theory means that the higher the luminance level, the higher the JND value. Because the proposed JND thresholds were detected at the intensity value of 128, for other intensity values, a modification factor should be included. This effect is called

the *luminance adaptation* effect.

However, for displays, the brightness and the pixel values have a non-linear relationship. To correct the non-linearity, *gamma correction* formulated as Eq. (5.42) has to be carried out,

$$L = cI^{\gamma} \tag{5.42}$$

where L is the brightness value in cd/m^2, I is the pixel intensity value and ranges from 0 to 255 for 8-bit images. γ is the correction parameter and $\gamma = 2.2$ for CRT display.

From Eq. (5.42), Eq. (5.43) can be deduced:

$$\Delta I = (\gamma c)^{-1}(L/c)^{1/\gamma-1}\Delta L \tag{5.43}$$

Then Eqs. (5.44) and (5.45) are derived:

$$Jnd_I = (\gamma c)^{-1}(L/c)^{1/\gamma-1}Jnd_L \tag{5.44}$$

$$Jnd_{I128} = (\gamma c)^{-1}(L_{128}/c)^{1/\gamma-1}Jnd_{L128} \tag{5.45}$$

where, Jnd_I and Jnd_L are the JND thresholds for the pixel intensity values and the corresponding brightness values, respectively. Jnd_{I128} is the JND value at intensity value of 128, whereas Jnd_{L128} is the corresponding brightness JND value. L_{128} is the brightness at intensity value of 128.

Eq. (5.46) is deduced from Eqs. (5.42), (5.44) and (5.45):

$$\begin{aligned} F_{lum} &= \frac{Jnd_I}{Jnd_{I128}} \\ &= \frac{(\gamma c)^{-1}(L/c)^{1/\gamma-1}Jnd_L}{(\gamma c)^{-1}(L_{128}/c)^{1/\gamma-1}Jnd_{L128}} \\ &= (\frac{L}{L_{128}})^{1/\gamma-1}(\frac{Jnd_L}{Jnd_{L128}}) \\ &= (\frac{cI^{\gamma}}{c128^{\gamma}})^{1/\gamma-1}(\frac{Jnd_L}{Jnd_{L128}}) \\ &= (\frac{I}{128})^{1-\gamma}(\frac{Jnd_L}{Jnd_{L128}}) \end{aligned} \tag{5.46}$$

According to the Weber-Fechner law, the factor (Jnd_L/Jnd_{L128}) is a monotonously increasing function. However, $((I/128)^{1-\gamma})$ is monotonously decreasing, which compensates the Weber-Fechner effect, resulting in a U-shape curve for the luminance adaptation factor. The factors at the lower and higher intensity regions are larger than those in the middle intensity region. Finally, an experimental formula of the luminance adaptation factor is shown as follows:

$$F_{lum} = \begin{cases} (60 - \bar{I})/150 + 1 & \bar{I} \leq 60 \\ 1 & 60 < \bar{I} < 170 \\ (\bar{I} - 170)/425 + 1 & \bar{I} \geq 170 \end{cases} \tag{5.47}$$

where $\bar{I}$ is the average intensity value of the block.

5.3.4 Contrast Masking Based on Block Classification

Contrast masking is an important phenomenon in the HVS perception and is referred to as the reduction in the visibility of one visual component in the presence of another. Usually noise is less visible in the regions where texture energy is high, whilst noise is easily observed in the smooth and edge areas. Here, an accurate block classification method is proposed based on the Canny operator [Canny, 1986], where the blocks can be classified into three categories, namely Plane, Edge and Texture, respectively.

As we know, Canny operator is a very famous and powerful edge detector, which can detect the edge pixels accurately for a given image. The pictures in the middle column of Figure 5.6 show the edge maps detected by the Canny detector. For a given block, if it contains very sparse edge pixels, it can be considered as a smooth block. One the other hand, if it contains many edge pixels, it means that this block has a lot of high frequency energy and can be considered as a texture block. Thus, the blocks can be classified according to the density of edge pixels.

Based on the above analysis, a parameter named edge pixels density ρ_{edgel} is defined:

$$\rho_{edgel} = \Sigma_{edgel}/N^2 \tag{5.48}$$

where Σ_{edgel} is the number of edge pixels in a given block with the edge map generated by the Canny operator. The block type is determined by Eq. (5.49). In our experiments, it is found that $\alpha = 0.1$ and $\beta = 0.2$ work well for most images we have tried.

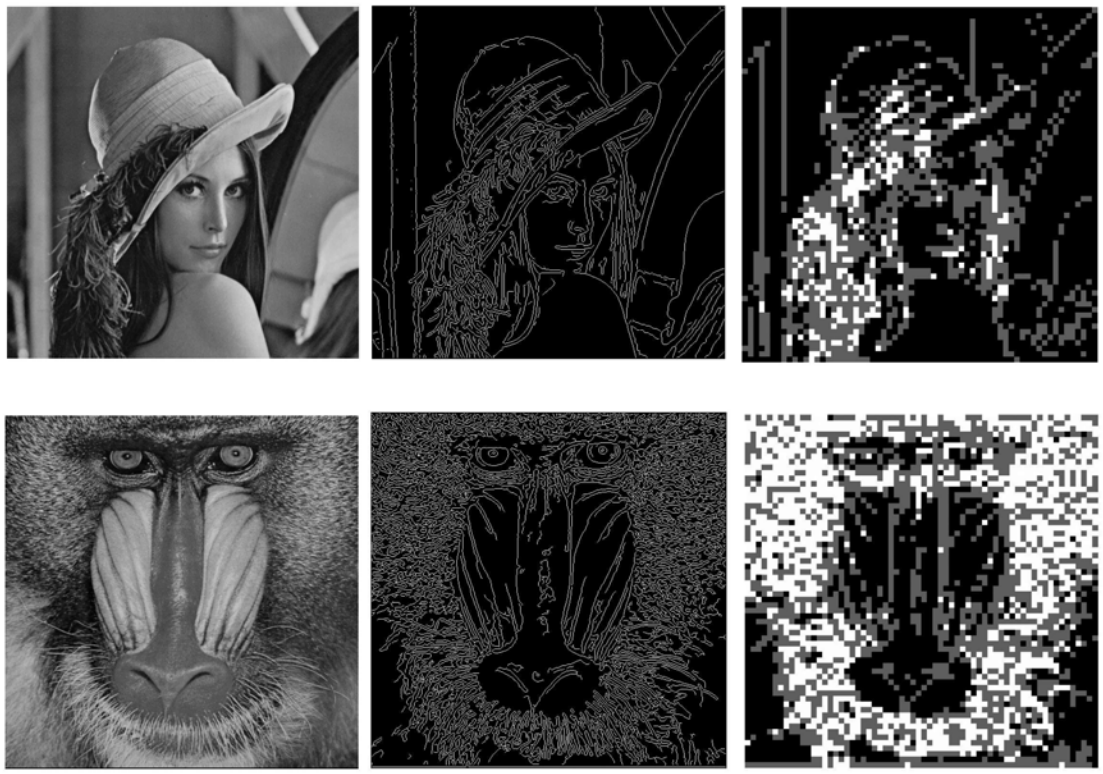

Figure 5.6: *Block classification results,where black represents Plane, grey represents Edge, white represents Texture.*

Figure 5.6 shows two examples of the block classification method.

$$Blocktype = \begin{cases} Plane & \rho_{edgel} \leq \alpha \\ Edge & \alpha < \rho_{edgel} \leq \beta \\ Texture & \rho_{edgel} > \beta \end{cases} \tag{5.49}$$

Human eyes are usually very sensitive to the distortion in the smooth area or around the edge, so the information should be protected in the smooth and edge blocks. For the texture area, eyes are less sensitive to the low frequency distortion, such as blocky artifacts, but the high frequency information should be preserved. Based on the above considerations, an elevation factor for each block type is determined by Eq. (5.50).

$$\Psi = \begin{cases} 1 & \text{for Plane and Edge block} \\ 2.25 & \text{for } (i^2 + j^2) \leq 16 \text{ subband in Texture block} \\ 1.25 & \text{for } (i^2 + j^2) > 16 \text{ subband in Texture block} \end{cases} \tag{5.50}$$

where i and j are DCT subband indices.

Considering the intra-band masking effect [Zhang et al., 2005; Hontsch and Karam, 2002; Foley and Boynton, 1994], the masking factor is finally obtained:

$$F_{contrast}(n,i,j) = \begin{cases} \Psi, & \text{for } (i^2+j^2) \leq 16 \text{ in Plane and Edge block} \\ \Psi \cdot min(4, max(1, (\frac{C(n,i,j)}{T_{Basic(n,i,j)\times F_{lum}(n)}})^{0.36}), & other \end{cases} \tag{5.51}$$

Then the modulation factor F_M in Eq. (5.29) is obtained by the product of F_{lum} and $F_{contrast}$ as shown in Eq. (5.30), completing the derivation of the spatial JND profile.

5.4 Temporal JND Model

5.4.1 Temporal Modulation Factor

In considering the temporal effect, the temporal modulation factor needs to be evaluated. Final JND map is generated by Eq. (5.28).

Let $CSF(f_s, f_t)$ denote the spatio-temporal contrast sensitivity function. It has been shown [van den Branden Lambrecht and Kunt, 1998; Girod, 1993; Daly, 1998] that there is no separable characteristic between the spatial and temporal frequency. It means that $CSF(f_s, f_t) \neq CSF_S(f_s)CSF_T(f_t)$. Thus, the temporal modulation factor F_T not only relies on the temporal frequency, but also depends on the spatial frequency.

Robson [Robson, 1966] gave the temporal contrast sensitivity experimental data as shown in Figure 5.7. It is easily found that although the temporal CSF is related to the spatial frequency at lower spatial frequencies, it seems to follow a similar shape at the higher spatial frequencies. Even for low spatial frequencies, at higher temporal frequencies (larger than 10 Hz), the curves also show this shape. The logarithm of the contrast sensitivity values are plotted for the spatial frequencies of 4 cpd, 16 cpd and 22 cpd in Figure 5.8. It is observed that the three curves have almost the same slope. The slope by curve-fitting equals to -0.0558. It means that this shape satisfies the exponential function. However, the JND threshold is overestimated in the experiments if -0.0558 is adopted. By considering the tradeoff between the perceived quality and the ability to conceal distortion, an empirical slope -0.03 is used in Eq. (5.52). The conclusion could be drawn that for higher spatial frequency, or higher temporal

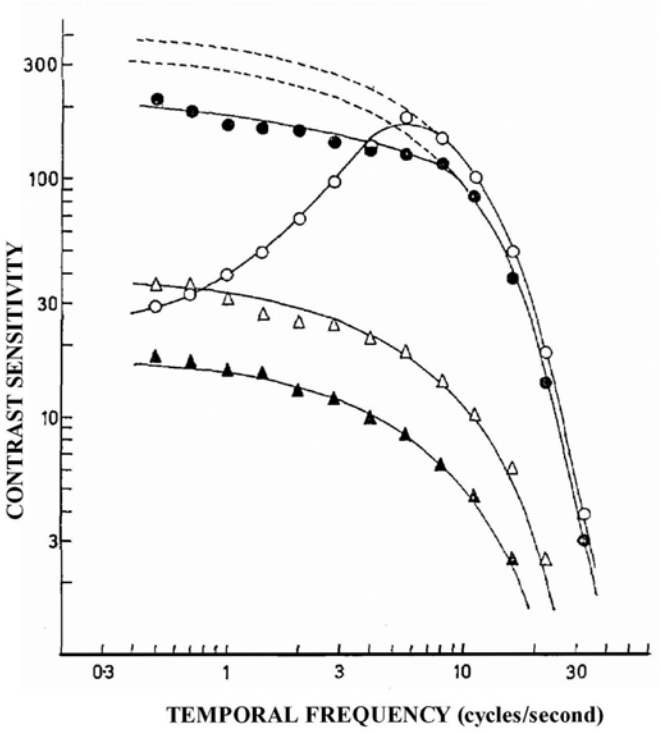

Figure 5.7: *Temporal CSF curve for different spatial frequency: 0.5 cpd (open circle), 4 cpd (filled circle), 16 cpd (open triangle), 22 cpd (filled triangle) [Robson, 1966]*

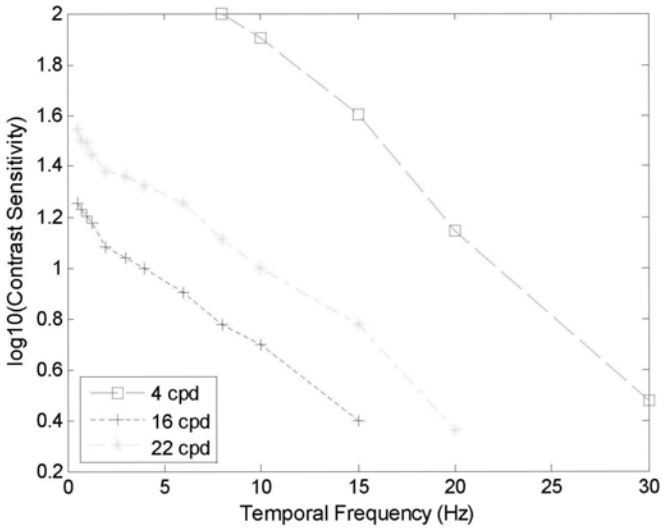

Figure 5.8: *Logarithmic function of the temporal CSF*

frequency at lower spatial frequency, the temporal contrast sensitivity is only related to the temporal frequency. Thus,

$$10^{-0.03\times f_t} = CSF(f_t)/CSF(f_{t0}) \tag{5.52}$$

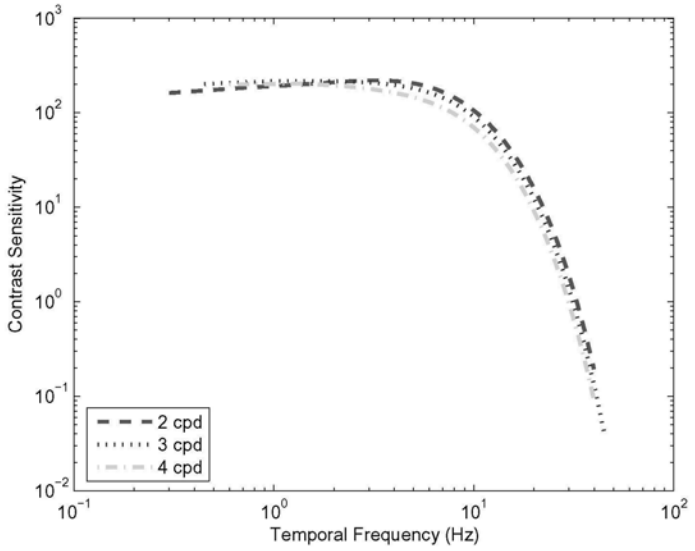

Figure 5.9: *Temporal CSF for low spatial frequency*

where, f_t is temporal frequency in Hz. f_{t0} means $f_t = 0$. Since the JND model is the reciprocal of CSF value and $1/CSF(f_{t0})$ is just the spatial JND value, then Eq. (5.53) can be deduced. In this case, the temporal modulation factor $F_T = 1.07^{f_t}$, which gives

$$T_{JND} = T_{JND_s} \times 1.07^{f_t} \tag{5.53}$$

The CSF has the characteristic of a band-pass filter at the lower spatial frequencies as shown in Figure 5.7. However, at the usual viewing distance, the spatial frequencies of (0,1) and (1,0) AC coefficients are larger than 2 cpd, which are the lowest frequency components in DCT domain. According to the Kelly's model [Kelly, 1979], the contrast sensitivity curve is given in Figure 5.9. It could be found that the contrast sensitivity is nearly constant for the temporal frequencies less than 10 Hz. In this case, the temporal modulation factor $F_T = 1$. From the discussion above, the formula for calculating temporal modulation factor F_T is derived:

$$F_T = \begin{cases} 1 & f_s < 5\,cpd\ \&\ f_t < 10Hz \\ 1.07^{(f_t - 10)} & f_s < 5\,cpd\ \&\ f_t \geq 10Hz \\ 1.07^{f_t} & f_s \geq 5\,cpd \end{cases} \tag{5.54}$$

5.4.2 How to Compute Temporal Frequency

In this section, the computation of the temporal frequency will be explained. The temporal frequency of a video signal depends on the rate at which the image varies. It depends not only on the motion, but also on the spatial frequency of the object. In [Wang et al., 2002], it was proved that:

$$f_t = f_{sx} v_x + f_{sy} v_y \tag{5.55}$$

where f_{sx} and f_{sy} are horizontal and vertical component of the spatial frequency, respectively. (v_x, v_y) is the velocity of object motion on the retina plane in degree/s. For a DCT frequency component located in i_{th} column and j_{th} row, f_{sx} and f_{sy} can be calculated by Eq. (5.56), where N=8, θ_x and θ_y are the horizontal and vertical visual angles obtained by Eq. (5.35).

$$f_{sx} = \frac{i}{2N\theta_x} \qquad f_{sy} = \frac{j}{2N\theta_y} \tag{5.56}$$

From Eq. (5.55), it also can be found that not only the magnitude of motion, but also the directionality can influence the temporal frequency. For example, for two motion vectors (3, 4) and (5, 0), although they have same magnitude 5, they will produce different temporal effects for a 2-D spatial frequency.

Human eyes can automatically move to track an observed object. This phenomenon is called *smooth pursuit eye movement* (SPEM). SPEM can slow down the velocity of object motion projected on the retina. Besides the SPEM, another two types of eye movements are reported: the *natural drift eye movement* and the *saccadic eye movement*. The former is very slow (0.8∼1.5 deg/s) and the latter is related to rapid movement of the eyes.

In [Daly, 1998], the retinal image velocity can be calculated by Eq. (5.57):

$$v_\hbar = v_{I\hbar} - v_{E\hbar} \qquad (\hbar = x, y) \tag{5.57}$$

where $v_\hbar$ and $v_{I\hbar}$ are the velocities on the retina plane and the image plane, respectively. $v_{E\hbar}$ is the eye movement speed which is determined as:

$$v_{E\hbar} = min[g_{spem} \times v_{I\hbar} + v_{MIN}, v_{MAX}] \tag{5.58}$$

where g_{spem} is the gain of the smooth pursuit eye movements with the empirical value of 0.98. v_{MIN} is the minimum eye velocity due to the drift movement and the classical value is 0.15 deg/s. v_{MAX} is the maximum velocity of the eyes corresponding to the saccadic eye movement and the value is normally 80 deg/s [Daly, 1998]. The velocity on the image plane $v_{I\hbar}$ can be obtained by following formula:

$$v_{I\hbar} = f_{fr} \times MV_{\hbar} \times \theta_{\hbar} \qquad (\hbar = x, y) \tag{5.59}$$

where f_{fr} is the frame rate of video sequence. $MV_{\hbar}$ is the motion vector of each block, which can be obtained by the block-based motion estimation algorithm. $\theta_{\hbar}$ is the visual angle of a pixel obtained by Eq. (5.35).

Finally, the temporal frequency for each DCT coefficient can be calculated by using Eq. (5.55). Then, the temporal modulation factor F_T is obtained by Eq. (5.54).

5.5 Experimental Results

JND models can avoid generating values larger than the actual HVS thresholds. A better JND model should yield larger JND values at a fixed perceived quality. To evaluate the performance of the JND models, noise is added to each DCT coefficient in an image/video according to the JND values generated by the JND profiles.

$$C'(k, n, i, j) = C(k, n, i, j) + f \times T_{JND}(k, n, i, j) \tag{5.60}$$

where $C(k, n, i, j)$ is the the $(i, j)_{th}$ DCT coefficient in the n_{th} block of k_{th} frame. f takes the value of +1 or -1 randomly, to avoid introducing a fixed pattern of changes. At the same perceptual quality, the higher the injected-noise energy (measured by PSNR), the more accurate is the JND model. It means that the JND model can tolerate more distortion at a given quality level.

5.5.1 Evaluation on Images

In this experiment, ten images were chosen for testing. Five are 512×512 images and the others are larger size images selected from some standard 720p HD sequences. Yang's JND profile [Yang et al., 2005b] was implemented and compared with the proposed JND model. Moreover, two DCT-based JND models were also implemented for more

Figure 5.10: *Noise-contaminated Barbara images: (a) Yang's model; (b) DCTune; (c) Zhang's model; (d) Proposed model*

convincing comparison. One is the classic JND model named DCTune [Watson, 1993], another is a more recent DCT-based JND model proposed by Zhang [Zhang et al., 2005].

The PSNR is used to measure the capability of distortion tolerance of the JND models. At the same perceived quality, a better model will achieve higher JND thresholds and result in lower PSNR. Table 5.1 shows the PSNRs of the four JND profiles. It can be found that the proposed method can tolerate more distortion among four models. According to the analysis for Eq. (5.36) in Section 5.3, with the increase in image size, the JND value for each DCT coefficient increases and more distortion can be concealed. The experimental results are consistent with the theoretical analysis and the three DCT-based models all follow this property. However, the performance

512×512				
Image	Yang's	DCTune	Zhang's	Proposed
Baboon	30.97	28.61	29.63	28.38
Barbara	30.83	29.76	30.29	29.50
Lena	31.90	30.95	31.16	29.97
Man	31.04	30.54	30.47	29.75
Pepper	31.01	30.79	30.53	29.99
Average	31.15	30.13	30.42	29.52
1280×720				
Image	Yang's	DCTune	Zhang's	Proposed
Night	29.48	28.43	24.70	24.81
Optis	33.43	27.44	26.65	24.72
Sailormen	30.72	27.26	25.29	24.75
Sheriff	30.78	28.39	25.59	24.82
Spincalendar	30.46	26.69	24.90	24.68
Average	30.97	27.64	25.43	24.76

Table 5.1*: PSNR compared with original image (in dB)*

of Yang's method is almost the same for both the small and large images, because it is a pixel-wise JND profile and is not based on the CSF. Therefore, the DCT-based methods agree with HVS more.

Figure 5.10 shows the noise-contaminated *Barbara* images created by four JND models. The Yang's model and DCTune model have obvious distortion in dark areas and around the boundaries of objects. The injected noise in Zhang's model and the proposed model is almost invisible. For a more convincing evaluation, the subjective viewing tests were conducted based on "Adjectival categorical judgement methods" recommended by ITU-R BT.500-11 standard [ITU-R, 2002]. In each test, two images were juxtaposed on the screen (left is the noise-injected image and right is the original image). Twenty subjects (ten are experts in image processing field and ten are naive) were asked to give quantitative scores for all the image pairs, using the continuous quality comparison scale shown in Table 5.2. In this experiment, the test equipment is a Viewsonic professional series P225fb CRT display (as specified in Section 5.3). The viewing distance is four times of the image height.

Table 5.3 shows the subjective scores and lower score means better subjective quality. Yang's method overestimates the JND thresholds for the lower luminance values, so the injected noise is still slightly visible in the images which have a lot of dark areas, such as Night and Sheriff. DCTune shows a little worse quality because the luminance

Subjective score	Description
-3	The right one is much worse than the left one
-2	The right one is worse than the left one
-1	The right one is slightly worse than the left one
0	The right one has same quality as the left one
1	The right one is slightly better than the left one
2	The right one is better than the left one
3	The right one is much better than the left one

Table 5.2: *Comparison scale for subjective quality evaluation*

Image		Yang's	DCTune	Zhang's	Proposed
512 × 512	Baboon	0.40	0.55	0.20	0.30
	Barbara	1.55	1.80	0.40	-0.05
	Lena	1.40	1.85	0.30	0.30
	Man	0.80	1.10	0.10	0.00
	Peppers	0.95	1.10	0.20	0.20
1280 × 720	Night	1.40	1.30	0.90	0.05
	Optics	0.70	1.10	0.70	0.20
	Sailorman	0.85	1.65	0.75	0.55
	Sheriff	1.40	1.50	0.60	0.35
	Spincalendar	0.96	2.10	1.0	0.70
Average		1.04	1.41	0.52	0.26

Table 5.3: *The subjective quality evaluation results*

masking and the contrast masking adopted in this model are not so accurate. Zhang's model has a better quality. The slightly higher quality score of Zhang's model compared with the proposed model is due to the excessive intra-band masking for the low frequency components in smooth blocks. The noisy images contaminated by the proposed JND profile have similar quality with the original ones. Average score is only 0.26. It means that the distortion introduced by the proposed method is almost invisible.

From the experimental results, it is concluded that the proposed JND profile not only conceals much more distortion, but also achieves better subjective quality than other methods. The proposed profile has a much better correlation with the HVS since it exploits the HVS efficiently and accurately.

5.5.2 Evaluation on Video Sequences

In this experiment, five SD (720×576) sequences and five HD (1280×720) sequences were chosen as test sequences. They all have 300 frames. SD and HD sequences were

SD (720x576) @ 25 fps								
Seq	Yang's		DCTune		Zhang's		Proposed	
	1_{st} frame	All	1_{st} frame	All	1_{st} frame	All	1_{st} frame	All
City	32.30	33.95	30.41	30.81	31.46	31.57	29.37	27.87
Crew	31.46	33.02	33.03	33.07	31.64	31.52	29.99	28.59
Harbour	31.92	33.87	29.23	29.16	31.18	31.16	29.49	27.51
Ice	31.78	34.60	31.62	31.48	30.87	31.16	29.96	28.22
Soccer	31.59	34.50	30.11	30.94	30.62	31.52	29.06	28.14
Average	31.81	33.99	30.88	31.09	31.15	31.39	29.57	28.07

Table 5.4: *PSNR compared with original SD videos (in dB)*

HD (1280x720) @ 60fps								
Seq	Yang's		DCTune		Zhang's		Proposed	
	1_{st} frame	All	1_{st} frame	All	1_{st} frame	All	1_{st} frame	All
Crew	32.98	34.41	26.94	27.33	26.24	26.18	25.10	23.04
Harbour	32.27	34.12	25.72	25.65	25.69	25.65	24.76	22.25
Night	29.46	31.29	28.43	28.37	24.71	24.62	24.81	22.45
Optis	33.39	35.44	27.44	27.43	26.65	26.71	24.72	22.32
Sailormen	30.69	32.71	27.26	27.31	25.29	25.39	24.75	22.72
Average	31.76	33.59	27.16	27.22	25.72	25.71	24.83	22.56

Table 5.5: *PSNR compared with original HD videos (in dB)*

tested at 25 frame/second and 60 frame/second, respectively. Yang's model [Yang et al., 2005b], DCTune [Watson, 1993] and Zhang's model [Zhang et al., 2005] were also implemented for comparison.

Comparison for Distortion Tolerance Capability

From Table 5.4 and Table 5.5, it can be observed that the proposed JND profile can tolerate much more distortion than other profiles. In these two tables, the following phenomena could be found:

- In the proposed JND profile, the PSNRs of the HD sequences are lower than those of the SD sequences. One reason is that the picture size of HD is larger which decreases the visual angle. Therefore, the spatial frequency of each DCT component increases, resulting in larger JND value. Another reason is that the frame rate of HD sequences is higher, resulting in higher temporal frequency. According to the proposed temporal JND model, JND value will be larger. So, the HD sequences will tolerate more distortion than the SD sequences.

- In the proposed JND profile, the average PSNR of whole sequences is lower than that of the first frame. This is because there is no motion in the first frame and only the spatial JND profile works . According to the proposed JND model, the temporal modulation factor will only be introduced in the subsequent frames. Since the human eyes are usually not sensitive to the distortion in the large motion frames [Jia et al., 2006; Kelly, 1979; Wang et al., 2004], this result is consistent with the human visual system.

- In Yang's model, the PSNR of the subsequent frames is higher than that of the first frame. This is because the temporal masking factor used in this model is based on the inter-frame difference. The empirical curve is shown in Figure 5.2. The minimum scaling factor is 0.8. Since the inter-frame difference is not high in most cases, thus the scaling factors are usually less than 1. Therefore, the temporal JND model in [Yang et al., 2005b] will reduce the JND value and results in higher PSNRs for the subsequent frames.

- In DCTune and Zhang's model, the PSNRs are not much different between the first frame and the subsequent frames. The reason is that no temporal effects are introduced in these models.

Perceptual Quality Comparison

In order to test the perceptual quality of the proposed JND profiles, Double stimulus continuous quality scale (DSCQS) method, as specified in ITU-R BT.500 [ITU-R, 2002], was used to evaluated the video quality. Figure 5.11 shows the test scheme. The DSCQS method presents two videos A and B (twice each) to the viewers, where one is a source sequence and the other is a processed sequence. Twenty viewers (ten are experts in image processing field and ten are naive) were involved in experiments. The Mean Opinion Score (MOS) scales for the viewers to vote for the quality are: Excellent (100-80), Good (80-60), Fair (60-40), Poor (40-20), and Bad (20-0). In this experiment, the test equipment is still the Viewsonic professional series P225fb CRT display. The view distance is four times of the image height.

Difference mean opinion scores (DMOS) are calculated as the difference of MOSs between the original videos and the noise-injected videos. Smaller the DMOS means a

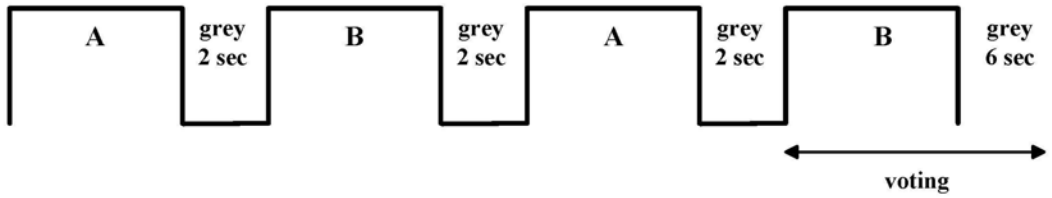

Figure 5.11: *DSCQS test scheme*

SD (720×576) @ 25 fps				
Seq	Yang's	Dctune	Zhang's	Proposed
City	15.8	19.3	11.0	12.7
Crew	18.0	17.7	5.3	5.7
Harbour	8.0	20.0	2.8	3.4
Ice	10.0	17.2	3.8	3.7
Soccer	10.0	14.2	6.8	0.4
Average	12.3	17.7	5.9	5.2
HD (1280×720) @ 60fps				
Seq	Yang's	Dctune	Zhang's	Proposed
Crew	6.9	13.3	4.0	5.8
Harbour	6.1	16.5	3.8	4.4
Night	9.5	13.3	6.8	5.9
Optis	4.3	10.3	3.0	3.6
Sailormen	7.1	10.3	3.8	3.1
Average	7.1	12.7	4.3	4.6

Table 5.6: *DMOSs for noise-injected videos @ CRT display*

higher perceptual quality of the processed video. Table 5.6 shows the averaged DMOSs of the twenty viewers for all the test sequences. The proposed JND profile has very small DMOS scores which are only 5.2 for the SD video and 4.6 for the HD video averagely. It means the videos with the distortion injected by the proposed JND model look almost the same as the original videos. Although Zhang's method has similar perceived quality compared with our proposed model, it cannot tolerate as much JND noise as our method, as shown in Tables 5.4 and 5.5.

5.6 Summary

In this chapter, a spatio-temporal JND profile is proposed. This model estimates the explicit JND threshold in the DCT domain. It is based on the spatial CSF, the luminance adaption effect compensated by gamma correction and the block classification based contrast masking effect. In order to exploit the temporal properties existing in

the videos, a temporal JND modulation factor is incorporated, which not only includes the temporal CSF effect, but also considers the retina motion compensation and the directionality of the motion. The proposed JND model was also evaluated in HD size images and videos. Experimental results have demonstrated the reliability of the proposed JND profile. Compared with other models, the proposed model can tolerate more distortion and has better perceptual quality.

Since the proposed JND model is DCT-based, it can be easily applied to many image and video processing algorithms, such as JPEG, MPEG1/2/4, H.261/263/264, AVS, and so on.

Part of the work in this chapter was presented at *ICIP2008*, entitled "A Temporal Just-noticeable Distortion Profile for Video in DCT Domain ", and *ICME2008*, entitled " Spatial Just Noticeable Distortion Profile for Image in DCT Domain ", respectively. All the work in this chapter was accepted for publication in *IEEE Transactions on Circuits and System for Video Technology* as a regular paper, entitled "Spatio-temporal Just Noticeable Distortion Profile for Grey Scale Image/Video in DCT Domain".

Chapter 6

JND Based Perceptual Picture Coding Techniques

6.1 Introduction

The image and video compression techniques have developed very fast in past two decades. Driven by the growing requirement of storage and transmission of visual information, lots of international image/video compression standards appeared and enriched our life. The state-of-the-art JPEG2000 and H.264/MPEG-4 AVC are two newest standards that exceed their previous generations in terms of coding efficiency. However, these compression methods are all signal processing based and share a common framework which is predictive transform followed by entropy coding. With the development of over two decades, it has become more and more difficult to improve the coding performance. In order to increase the compression efficiency, more and more complicated techniques are introduced in compression standards, such as defining a number of "modes" to deal with different kinds of image or video regions, utilizing exhausting search to achieve rate-distortion optimization, or designing memory-consuming entropy coding techniques to handle different kinds of correlations. Consequently, small improvements are accomplished with the great expense of increasing complexity at both encoder and decoder. When looking back the development of the image and video coding standards, it is obvious that the traditional image and video compression are meeting the bottleneck.

Another fundamental problem in current mainstream compression schemes is that only the statistical redundancy among pixels is considered as the objective of the optimization, but the perceptual redundancy is almost neglected. Thus, the pixel-wise distortion metric, such as mean squared error (MSE) and peak signal-to-noise ratio (PSNR), are widely adopted in the traditional coding standards. But unfortunately,

they have been widely criticized for not correlating well with perceived quality measurement [Girod, 1993]. The removal of the perceptual redundancy has many advantages. Firstly, it ensures that only the visually important information is encoded. Secondly, better compression performance can be achieved by discarding perceptually unnecessary information. It is very important to develop the new compression techniques by considering the perceptual redundancy. Perceptual optimized compression seems to open a bright window for the future image and video compression. In recent years, perceptual redundancy has already been considered in several works. There are two new trends emerging in non-traditional picture compression to exploit the perceptual redundancy existing in the pictures.

The first direction is inspired by the remarkable progress in computer vision as well as computer graphics. There are a lot of advanced techniques which would benefit the image and video compression towards improving the perceptual quality. Among the various vision techniques, image inpainting and texture synthesis are two promising methods to be utilized in image compression. Image inpainting, also called image completion, has potential to bring profit for the image compression. This technique is utilized to fill-in missing data in more general regions of an image in a visually plausible way. Several compression schemes have been presented in literatures which incorporate the image inpainting to remove the perceptual redundancy. In [Rane et al., 2003], some smooth regions are directly removed and restored at the decoder side by using image inpanting. In [Liu et al.,], edge is extracted as the auxiliary information to guide the inpainting process in order to keep the perceptual quality in the edge involved regions. In this method, not only smooth regions, but also some unnecessary structural regions can be skipped, thus more visual redundancy is reduced. Xiong et al. [Xiong et al., 2007] proposed a gradient-based inpainting method to compress the image with large-scale graduation regions where the hue and lightness changes smoothly. Texture synthesis is an alternative way to generate a large size texture region from a given finite texture sample. It is very effective to process the texture regions. These computer vision based non-traditional picture compression methods will be discussed in the Chapter 7 thoroughly.

Another trend is to incorporate human visual system (HVS) into the compression schemes. Since human eyes are the ultimate receiver of the images and videos, it

is very important and advantageous to incorporate HVS into the image and video compression algorithms. In these HVS based picture compression schemes, how to model HVS accurately and efficiently is a vital problem. JND can model HVS very well, which refers to the maximum distortion which cannot be perceived by the human eyes. Knowledge on JND can guide the image processing algorithms to remove the perceptual redundancy. In Chapter 5, a DCT-based spatio-temporal JND profile is proposed. This model incorporates the spatial contrast sensitivity function (CSF), the luminance adaptation effect, and the contrast masking effect based on block classification. Gamma correction is also considered to compensate the original luminance adaptation effect to obtain more accurate results. Moreover, a temporal modulation factor is included by incorporating the temporal CSF and the eye movement compensation.

Based on this JND model, an improved transparent image coding algorithm is proposed, where the quantization factor can be tuned for each block according to the JND threshold. The experimental results show that the images compressed by the proposed method are hardly distinguished from the original images. Therefore, the proposed method can achieve perceptually transparent quality for images. Moreover, the bit-rate of the proposed algorithm is less than that of the state-of-the-art lossless and near-lossless codecs.

The proposed JND model is also applied in video coding. Based on the JND model, a perceptually optimized H.264 video coder is implemented. Benefited by the advantages of H.264 and the JND model, not only the statistical redundancy, but also the perceptual redundancy can be removed efficiently. As demonstrated in extensive experiments, compared with the original H.264 encoder, the perceptually transparent encoder can reduce bit-rate significantly with the same perceived quality.

The rest of the chapter is organized as follows. Section 6.2 introduces the proposed perceptually transparent image coding algorithm. The proposed perceptually optimized H.264 video coder is presented in Section 6.3. Finally, this chapter is summarized in Section 6.4.

6.2 The Perceptually Transparent Coding for Image

Image coding or image compression algorithms are used to obtain compact digital representation of image signals for the purpose of efficient transmission or storage. Image

compression can be classified into two categories, lossy and lossless coding. Lossless coding can make the processed image totally identical with the original and has many applications. Lossless compression methods usually are preferred for high-value content, such as medical images or scanned images made for archival purposes. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. So far, a lot of lossless image compression methods were proposed, such as JPEG-LS [Weinberger et al., 1996], PNG [ISO/IEC, 2004b], etc. Some international image coding standards also support lossless compression, including JPEG 2000 [Taubman and Marcellin, 2002] and JBIG2 [ITU-T and ISO/IEC, 2001]. Most of these main-stream techniques are signal-processing-based and treat images as 2-D signals. In these methods, only statistical redundancy among pixels are exploited, but the perceptual features are often neglected, so the compression ratios of these lossless coding methods are not very high. Since the human visual system (HVS) is the ultimate receiver of the majority of processed images, it is very important and advantageous to incorporate HVS into the image processing algorithms.

The concept of the transparent coding first appears in audio processing field [Painter and Spanias, 2000; Sinha and Tewfik, 1993]. It means that the generating output audio cannot be distinguished from the original input, even by a sensitive listener. If we extend this concept to the image compression, one straightforward idea emerges: could we achieve a perceptually lossless image compression algorithm by removing not only the statistical redundancy, but also the perceptual redundancy? In order to solve this problem, how to model the perceptual redundancy existing in image becomes a very important research issue. Fortunately, just-noticeable distortion (JND) gives us a good solution.

JND refers to the maximum distortion which cannot be perceived by the human eyes. Knowledge on JND can guide the image processing algorithms to remove the perceptual redundancy. JND model can be easily applied in many related areas, such as compression, watermarking, error protection, perceptual distortion metric, and so on. The computational JND model has been studied for a long time [Chou and Li, 1995; Chou and Chen, 1996; Chin and Berger, 1999; Yang et al., 2005b; Watson, 1993; Watson et al., 1997; Zhang et al., 2005; Ahumada and Peterson, 1992; Hontsch and Karam, 2002; Wei and Ngan, 2008b; Wei and Ngan, 2008c]. Generally, the existing

JND models belong to two categories. One is the model produced in image domain which is also called pixel-wise JND model [Chou and Li, 1995; Chou and Chen, 1996; Chin and Berger, 1999; Yang et al., 2005b]. Another one is named subband JND model which is determined in transform domain [Watson, 1993; Watson et al., 1997; Zhang et al., 2005; Ahumada and Peterson, 1992; Hontsch and Karam, 2002]. Because most of image and video compression schemes are performed in DCT domain, the DCT-based JND model attracts the interest of many researchers. In [Wei and Ngan, 2008b] [Wei and Ngan, 2008c], we proposed a DCT-based JND profile. This model estimates the explicit JND threshold in DCT domain. It not only incorporates the spatial contrast sensitivity function (CSF), but also considers all major vision effects, i.e., the luminance adaption and the contrast masking. This model is very easy to be applied in the image compression.

Based on our proposed JND model, a perceptually transparent image compression method for monochromatic images is presented, where the quantization factor can be tuned for each block according to the JND threshold. This makes the quantization error to be lower than the JND threshold. The proposed algorithm is also extended to the color images. The experimental results show that the images compressed by the proposed method are hardly distinguished from the original images. Therefore, the proposed method can achieve perceptually transparent quality for images. Moreover, the proposed algorithm consumes much less bits than other state-of-the-art lossless and near-lossless codecs.

6.2.1 The Spatial Just-noticeable Distortion Model

As described in Section 5.3, the spatial just-noticeable distortion threshold in DCT domain is typically expressed as a product of a base threshold and a modulation factor. n is the index of the block, i and j are the DCT coefficients' indices (i, $j = 0$ to 7 in our case). Then the corresponding JND can be expressed as:

$$T_{JND_s}(n,i,j) = T_{Basic}(n,i,j) \times F_M(n,i,j) \tag{6.1}$$

$$F_M(n,i,j) = F_{lum}(n) \times F_{contrast}(n,i,j) \tag{6.2}$$

where T_{JND_s} is the spatial JND threshold. T_{Basic} is the base threshold which is generated by the spatial contrast sensitivity function (CSF). The detailed equation is

presented in 5.3.

The modulation factor $F_M(n,i,j)$ is the product of the luminance adaptation factor F_{lum} and the contrast masking factor $F_{contrast}$, and F_{lum} is defined as follows:

$$F_{lum} = \begin{cases} (60 - \bar{I})/150 + 1 & \bar{I} \leq 60 \\ 1 & 60 < \bar{I} < 170 \\ (\bar{I} - 170)/425 + 1 & \bar{I} \geq 170 \end{cases} \tag{6.3}$$

where $\bar{I}$ is the average intensity value of the block.

For $F_{contrast}$, it is obtained by the following equations:

$$\Psi = \begin{cases} 1 & \text{for Plane and Edge block} \\ 2.25 & \text{for } (i^2 + j^2) \leq 16 \text{ in Texture block} \\ 1.25 & \text{for } (i^2 + j^2) > 16 \text{ in Texture block} \end{cases} \tag{6.4}$$

where i and j are DCT subband indices.

Considering intra-band masking effect [Zhang et al., 2005; Hontsch and Karam, 2002; Foley and Boynton, 1994], we finally obtain the masking factor:

$$F_{contrast}(n,i,j) = \begin{cases} \Psi, & \text{for } (i^2 + j^2) \leq 16 \text{ in Plane and Edge block} \\ \Psi \cdot min(4, max(1, (\frac{C(n,i,j)}{T_{Basic(n,i,j) \times F_{lum}(n)}})^{0.36}), & \text{otherwise} \end{cases} \tag{6.5}$$

6.2.2 Proposed Perceptually Transparent Coding Method

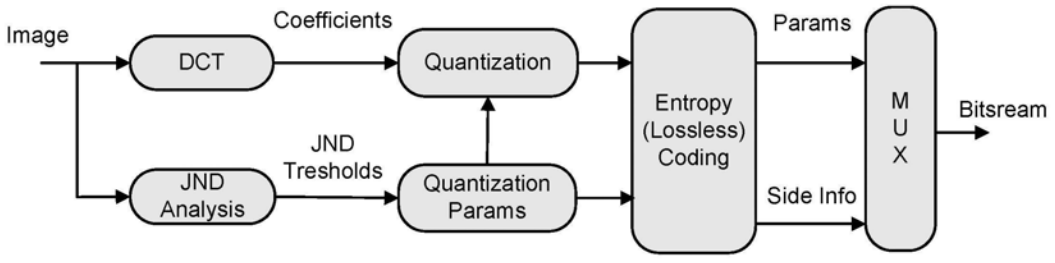

Figure 6.1: *The framework of proposed compression scheme*

The whole structure of our proposed compression scheme is depicted in Figure 6.1. Firstly, DCT transform and the JND analysis are performed for the input image. Secondly, the quantization steps are tuned by the JND thresholds to guarantee that the quantization error is lower than the JND thresholds. Finally, the quantized coefficients

and the side information are encoded by entropy coding and transmitted to the channel. For each DCT coefficient in a given block, the corresponding quantization step is computed as

$$Q_s(n,i,j) = \lfloor \alpha \times T_{JND_s}(n,i,j) \rfloor \tag{6.6}$$

where Q_s is the quantization step, $\lfloor x \rfloor$ is the floor function, which returns the highest integer less than or equal to x, α is a tuning factor that is chosen from 16 candidates whose range is from 2 to 6 as shown in Table 6.1. For a given block, α is a fixed number.

Index	α value	Index	α value
0	2	8	4.25
1	2.5	9	4.5
2	2.75	10	4.75
3	3	11	5
4	3.25	12	5.25
5	3.5	13	5.5
6	3.75	14	5.75
7	4	15	6

Table 6.1: *Tuning factor indices and values*

Since our JND model is based on many visual effects, such as luminance adaptation, contrast masking, etc., the JND thresholds for each block are different, which will result in different quantization matrix for each block. In order to restore the quantization matrix exactly for each block at the decoder side, the side information will consume lots of bits.

In order to design an efficient transparent image codec, the proposed JND model should be slightly modified. Thus, Eq. (6.6) is also changed to Eq. (6.7):

$$Q_s(n,i,j) = \left\lfloor \alpha \times \hat{T_{JND_s}}(n,i,j) \right\rfloor \tag{6.7}$$

where $\hat{T_{JND_s}}$ stands for the simplified JND threshold.

Modified JND model

In the modified JND model, $\hat{T_{JND_s}}$ is calculated by Eq. (6.8):

$$\hat{T_{JND_s}}(n,i,j) = T_{Basic}(n,i,j) \times F_{lum}(n) \times F_{contrast}(n,i,j) \tag{6.8}$$

where T_{Basic} is not changed, but F_{lum} and $F_{contrast}$ are modified by the following functions:

$$F_{lum} = \begin{cases} (60 - \hat{I})/150 + 1 & \hat{I} \leq 60 \\ 1 & 60 < \hat{I} < 170 \\ (\hat{I} - 170)/425 + 1 & \hat{I} \geq 170 \end{cases} \tag{6.9}$$

$$\hat{I} = (\bar{I} >> 1) << 1 \tag{6.10}$$

where $\bar{I}$ is the average intensity value of the block.

For $F_{contrast}$, the intra-band masking effect is ignored for simplicity, and we finally obtain the modified masking factor:

$$F_{contrast}(n, i, j) = \Psi \tag{6.11}$$

$$\Psi = \begin{cases} 1 & \text{for Plane and Edge block} \\ 2.25 & \text{for } (i^2 + j^2) \leq 16 \text{ in Texture block} \\ 1.25 & \text{for } (i^2 + j^2) > 16 \text{ in Texture block} \end{cases} \tag{6.12}$$

where i and j are DCT subband indices.

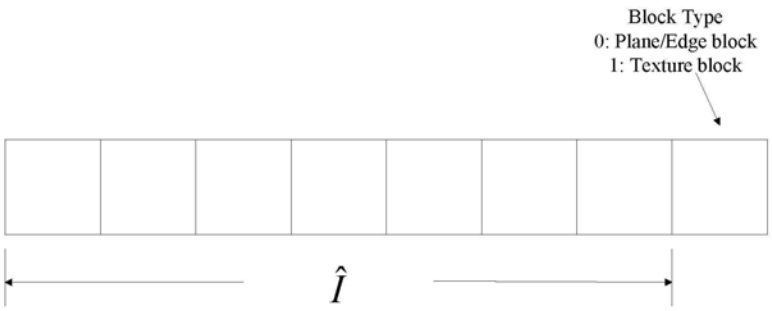

Figure 6.2: *Fixed length coding for the overhead*

Thus, for each block, JND thresholds are only related to $\hat{I}$ and the block type. In the proposed transparent codec, one byte is used to signal the $\hat{I}$ and the block type of each block as shown in Figure 6.2. The left seven bits are utilized to record the $\hat{I}$ and the last bit is used to signal the block type. 0 and 1 stand for plane/edge block and texture block, respectively.

How to determine the tuning factor α

In the proposed transparent image codec, tuning factor α still needs to be determined. If it is too small, although no perceptually noticeable errors are introduced, the perceptual redundancy may not be exploited thoroughly. However, if it is too large, the distortion may be noticed by the human eyes. Therefore, it is important to determine the value of α. The α is obtained by Eq. (6.13):

$$Max(\alpha) \quad \text{subject to} \quad D(n,i,j) < T_{JND_s}(n,i,j) + 1 \quad \text{for} \quad (i+j) \leq 8 \tag{6.13}$$

where D is quantization error obtained by Eq. (6.14). T_{JND_s} is the JND threshold computed by the original JND model as described in Section 6.2.1, i and j are the indices of the DCT coefficients.

$$D(n,i,j) = abs(C(n,i,j) - round(C(n,i,j)/Qs) \times Qs) \tag{6.14}$$

where *abs* and *round* stand for the absolute value function and the rounding operator, respectively. C is the DCT coefficient and Qs is the quantization step computed by Eq. (6.7).

As shown in Table 6.1, there are totally 16 tuning factor candidates. For a block, only 4 bits or equivalently 0.0625 bpp are consumed for encoding this information.

6.2.3 Extension to Color Images

Since most images perceived by human eyes are in color, it is necessary to extend the proposed perceptually transparent image coding method to color images. The difference between monochrome image and color image is that the latter one contains the chrominance information. If a color image is converted from RGB space to YC_bC_r space, the luminance component Y can be treated as a monochromatic image and processed by the proposed method described in Section 6.2.2. For the color difference components C_b and C_r, since the human visual system is less sensitive to the position and motion of color than luminance, bit-rate can be reduced by downsampling the chrominance components. At normal viewing distances, no perceptible distortion is introduced. Thus, the structure of the compression scheme for color image is depicted in Figure 6.3.

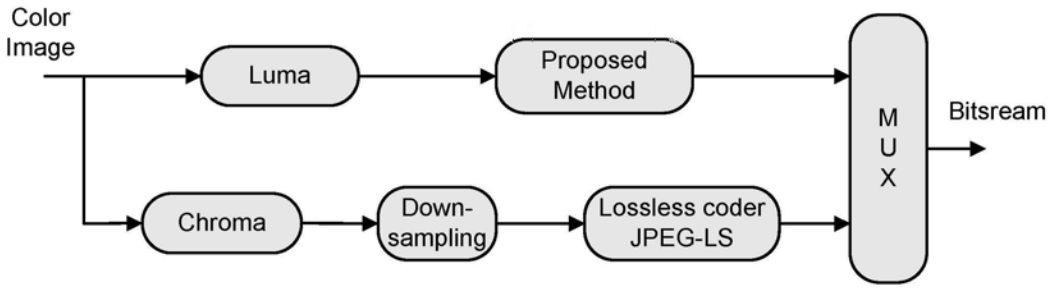

Figure 6.3: *The framework of the compression scheme for color image*

The input color image is firstly decomposed into luminance and chrominance components. The luma component is handled by the proposed perceptually transparent coding method. The chroma component is downsampled and encoded by lossless coding method JPEG-LS. Finally, the coded luma and chroma are combined together into one bitstream. Here, both C_r and C_b are subsampled 2:1 vertically and horizontally. The downsampling function is shown as follows:

$$C'_{\hbar}(m,n) = (C_{\hbar}(2m,2n) + C_{\hbar}(2m+1,2n) + C_{\hbar}(2m,2n+1) + C_{\hbar}(2m+1,2n+1) + 2)/4$$
$$(\hbar = b, r) \tag{6.15}$$

where $C'_{\hbar}$ and $C_{\hbar}$ stand for the downsampled and the original chroma components, respectively.

6.2.4 Experimental Results

In this experiment, our proposed method was evaluated on selected fourteen color images ranging from 512×512 to 1280×720. The proposed method was compared with JPEG2000 lossless coder (J2KL) [Taubman and Marcellin, 2002] and JPEG-LS coder [Weinberger et al., 1996] in terms of bit-rate. JPEG-LS is a state-of-the-art lossless coding algorithm for still images. Moreover, the JPEG-LS NLOCO near-lossless coder [Weinberger et al., 1996] with d=2 and d=9 was also implemented. The error, d, specifies the maximum pixel difference between the original image and the NLOCO compressed image. When d=2, the images compressed by the NLOCO coder are hardly seen to be different from the originals. The experimental results are shown in Table. 6.2. Compared with J2KL, JPEG-LS and NLOCO$_{d=2}$, the proposed method can save on average 74.6%, 77.9% and 56.3% of bitrate, respectively. When d=9, the bitrate of

Picture		J2KL	JPEG-LS	NLOCO$_{d=2}$	NLOCO$_{d=9}$	Proposed
512 × 512	Avion	11.54	11.84	5.72	2.28	2.78
	Butterfly	14.45	14.11	8.25	3.88	3.28
	Lena	13.59	13.60	6.98	3.19	2.99
	Milk	11.91	10.70	4.82	1.73	2.51
	Peppers	14.80	14.27	7.69	3.40	3.19
	Toucan	7.09	7.89	3.86	1.86	1.98
704 × 576	City	9.47	13.89	7.51	4.01	2.92
	Crew	8.92	10.68	5.05	2.29	2.31
	Harbour	9.48	12.94	6.74	3.98	2.94
	Ice	7.16	9.40	3.88	1.34	1.67
1280 × 720	Night	9.47	11.54	5.97	3.06	2.70
	Optis	7.63	10.70	5.15	2.18	2.05
	Sheriff	7.71	10.26	4.94	2.28	2.15
	Spincalendar	11.30	14.03	7.50	3.48	3.18
Average		10.32	11.85	6.00	2.78	2.62

Table 6.2: *Bit-rate comparison for color images (bpp)*

NLOCO is similar to that of the proposed coder. However, the distortion is visible in the images compressed by the NLOCO coder (as shown in Figure 6.4 and Figure 6.5).

The proposed perceptually transparent coding method was also evaluated on luminance component of the color images. The experimental results are shown in Table 6.3.

Figure 6.4: *Coding results for color images Butterfly and Toucan, from left to right: original, NLOCO$_{d=9}$, the proposed coder*

Picture		J2KL	JPEG-LS	NLOCO$_{d=2}$	Proposed
512 × 512	Avion	3.99	3.80	1.83	1.48
	Butterfly	5.03	4.78	2.62	1.84
	Lena	4.31	4.24	2.09	1.43
	Milk	3.77	3.63	1.66	1.18
	Peppers	4.62	4.51	2.29	1.51
	Toucan	2.73	2.63	1.30	1.06
704 × 576	City	4.61	4.63	2.50	1.97
	Crew	3.55	3.48	1.63	1.22
	Harbour	4.43	4.31	2.24	1.92
	Ice	3.30	3.12	1.26	1.06
1280 × 720	Night	3.89	3.78	1.94	1.53
	Optis	3.64	3.56	1.71	1.40
	Sheriff	3.49	3.40	1.63	1.80
	Spincalendar	4.70	4.62	2.44	1.37
Average		4.00	3.89	1.94	1.48

Table 6.3: *Bit-rate comparison for luminance component of the color images (bpp)*

Compared with J2KL, JPEG-LS and NLOCO$_{d=2}$, the proposed method can save on average 63.0%, 62.0% and 23.7% of bitrate, respectively.

For a more convincing evaluation, the subjective viewing tests were conducted based on "Adjectival categorical judgement methods" recommended by ITU-R BT.500-11 standard [ITU-R, 2002]. In this test, the original and the processed images were shown on the screen side by side. Fifteen subjects were asked to vote whether they were

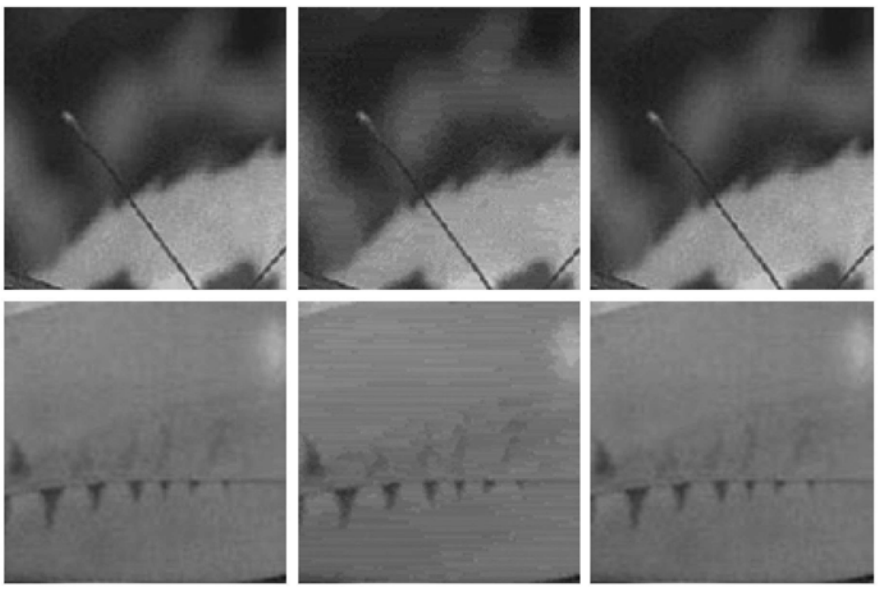

Figure 6.5: *Enlarged parts of Butterfly and Toucan, from left to right: original, NLOCO$_{d=9}$, the proposed coder. The obvious distortion can be seen in the images compressed NLOCO$_{d=9}$*

Picture	Ratio	Picture	Ratio
Avion	15 (100%)	Butterfly	14 (93%)
Lena	15 (100%)	Milk	15 (100%)
Peppers	15 (100%)	Toucan	14 (93%)
City	15 (100%)	Crew	13 (87%)
Harbour	15 (100%)	Ice	15(100%)
Night	14 (93%)	Optis	15 (100%)
Sheriff	14 (93%)	Spincalendar	14 (93%)
Total	202 (96%)		

Table 6.4*: Ratio of "Identical" votes for color images*

identical for each image pairs. In this experiment, the test equipment is a Viewsonic professional series P225fb CRT display. The experimental results are shown in Table 6.4. It can be observed that the proposed coder can achieve transparent quality compared with the original images.

6.3 Perceptual Video Coding Techniques for H.264

H.264 is the latest standard for moving picture coding [ITU-T and ISO/IEC, 2005]. In order to achieve outstanding coding performance, many advanced techniques are used, such as: intra mode decision, variable block size motion estimation, 1/4 resolution motion estimation, multiple reference frames, deblocking filter, integer cosine transform, CABAC and CAVLC entropy coding etc. It is shown that these techniques can provide nearly 50% bit rate reduction compared with other previous standards (e.g. half or less the bit rate of MPEG-2, H.263, or MPEG-4 Part 2). However, H.264 only exploits the statistical redundancy exising in the video frames efficiently, but ignores the perceptual features in videos. It is very important and necessary to propose a perceptual H.264 codec to remove not only the statistical redundancy, but also the perceptual redundancy. Based on the proposed spatio-temporal JND model, a JND based perceptual H.264 codec is presented in this section. Benefited by the advantages of H.264 and the JND model, the proposed perceptual encoder can save plenty of bits compared with the traditional H.264 encoder, and the perceived quality is kept high at the same time.

6.3.1 Proposed JND Based Perceptual H.264 Codec

Since the JND threshold refers to the maximum distortion which cannot be perceived by the human eyes, the proposed JND model can assist us to achieve a perceptually

transparent video codec. The transparent video codec can be implemented in H.264. Compared with the original H.264 encoder, the perceptually transparent encoder can reduce bit-rate significantly with the same visual quality. The whole process is illuminated in Figure 6.6.

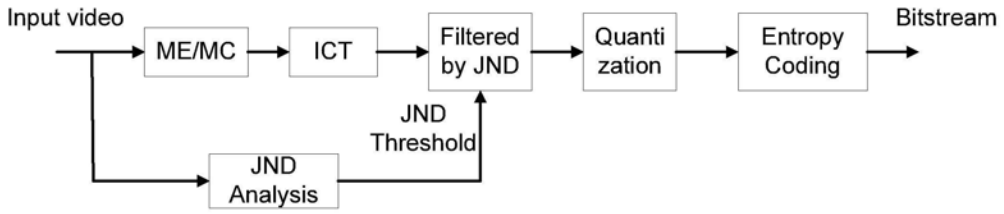

Figure 6.6: *The structure of the proposed perceptual H.264 encoder*

The procedure is described as follows:

Step1: The input video is processed by motion estimation and motion compensation first.

Step2: The residue field is performed by the integer cosine transform (ICT)

Step3: The ICT coefficients are filtered by JND model. The JND thresholds are calculated through the JND analysis. If the absolute value of the ICT coefficient is less than corresponding JND threshold, the ICT coefficient will be set as zero to save bit-rate. The filtering process is performed according to the following formula:

$$C'(k,n,i,j) = sign(C(k,n,i,j)) \times Max((C(k,n,i,j) - T_{JND}(k,n,i,j)),\ 0) \quad (6.16)$$

where C and C' are the original and the filtered DCT coefficient, respectively. k is the index of a frame in the video sequences and n is the index of a block in the k_{th} frame, and i and j are the DCT coefficients' indices. T_{JND} stands for the spatio-temporal JND threshold calculated by the proposed JND model in Section 5.3. *sign* stands for the sign function.

Step4: The filtered ICT coefficients are quantized and coded by entropy coding.

6.3.2 Experimental Results

Evaluate the proposed perceptual H.264 codec at the fixed QP

In this experiment, ten hight definition (HD) video sequences were chosen for the test. Five are *City*, *Crew*, *Harbour*, *Night*, *Sailormen* with the resolution of 1280×720 and at the frame rate pf 60 fps. They have 480 frames. Others are *Shields*, *Tractor*, *Riverbed*, *Stockholm*, *MobileCalendar* with the resolution of 1920×1080 and at the frame rate of 25 fps, which have a total of 200 frames. H.264 reference software JM 11.0 was compared with our proposed method. QP was set at 28. RD-optimization was turned on. Coding mode was IPPPPP. Table 6.5 shows the experimental results. The video sequences coded by the proposed coder have the similar visual quality to those coded by JM 11.0. The bit-rate saving ratio is 17% on average. For the *City*, *Calendar*, *Harbour* and *Stockholm*, bit-rate is reduced significantly. The reason is these squences contain a lot of texture and have higher JND values. Thus, more perceptual redudancy can be removed. But for *Tractor*, *Riverbed*, the bit saving ratio is no more than 10%, the reason is these squences are so smooth that the JND thresholds are not so high.

		JM 11.0			Proposed				
		Bit-rate (kb/s)	PSNR (db)	MOS	Bit-rate (kb/s)	PSNR (db)	MOS	Saving (%)	DMOS
1280 × 720 @ 60fps	City	5069.38	35.97	71.0	3542.97	34.85	70.7	30.11	0.3
	Crew	5633.78	38.83	69.1	4980.13	38.17	68.1	11.60	0.9
	Harbour	12376.01	36.45	69.7	10097.83	35.22	69.8	18.41	-0.1
	Cyclists	2248.04	40.15	69.1	2010.01	39.81	68.5	10.59	0.7
	Sailormen	8095.57	36.22	66.6	6586.04	35.28	66.8	18.65	-0.2
1920 × 1080 @ 25fps	Shields	6337.49	36.08	79.5	5358.96	34.83	77.9	15.44	1.5
	Tractor	7685.57	38.21	78.1	6991.77	37.45	78.1	9.03	0.0
	Riverbed	17823.51	38.05	76.3	16336.56	37.04	74.8	8.34	1.5
	Stockholm	4770.92	35.72	77.6	3824.27	34.60	78.5	19.84	-0.9
	Calendar	10607.26	35.59	82.1	7641.22	34.07	81.0	27.96	1.1
Average				73.9			73.4	17.00	0.5

Table 6.5: *Experimental results*

In order to test the perceptual quality of the proposed perceptual codec, double stimulus continuous quality scale (DSCQS) method, as specified in ITU-R BT.500 [ITU-R, 2002], was used to evaluate the video quality. Fifteen viewers were involved in the experiments. The Mean Opinion Score (MOS) scales for the viewers to vote for the quality are: Excellent (100-80), Good (80-60), Fair (60-40), Poor (40-20), and Bad (20-0). In this experiment, the test equipment is a Panasonic 65-inch plasma

display. Difference mean opinion scores (DMOS) are calculated as the difference of MOSs between the JM compressed videos and the videos compressed by the proposed encoder. Smaller DMOS means the difference of the perceptual quality between two video sequences is very small. Table 6.5 shows the averaged DMOSs of the fifteen viewers for all the test sequences. The average DMOS is only 0.5. It means the video sequences coded by the proposed coder have the similar visual quality to those coded by JM 11.0.

Evaluate the proposed perceptual H.264 codec at the same bit-rate

We also evaluated the proposed perceptual H.264 codec at the same bit-rate. The same ten HD size video sequences were chosen for testing. H.264 reference software JM 11.0 was compared with our proposed method. Rate control was turned on and the initial QP was set at 28. RD-optimization was turned on. Coding mode was IPPPPP.

		JM		Proposed		
		PSNR (db)	MOS	PSNR (db)	MOS	DMOS
1280 × 720 2M bps @60fps	City	33.42	60.2	33.21	64.3	4.2
	Crew	35.37	50.0	35.23	51.3	1.3
	Harbour	29.45	57.0	29.36	64.8	7.8
	Cyclists	39.78	64.0	39.77	63.7	-0.3
	Sailormen	32.01	56.8	31.90	56.3	0.5
1920 × 1080 2M bps @25fps	shields	32.25	61.2	32.15	69.5	8.3
	tractor	32.04	65.8	32.00	64.2	-1.7
	riverbed	28.68	46.7	28.71	46.0	-0.7
	Stockholm	33.34	66.3	33.05	74.7	8.3
	Calendar	31.21	57.0	31.01	62.8	5.8
Average			58.5		61.8	3.3

Table 6.6*: Experimental results*

In order to test the perceptual quality of the proposed perceptual codec, DSCQS method [ITU-R, 2002] was used to evaluated the video quality. Fifteen viewers were involved in experiments. The Mean Opinion Score (MOS) scales for the viewers to vote for the quality are: Excellent (100-80), Good (80-60), Fair (60-40), Poor (40-20), and Bad (20-0). In this experiment, the test equipment is still the Panasonic 65-inch plasma display. Difference mean opinion scores (DMOS) are calculated as the difference of MOSs between the JM compressed videos and the videos compressed by the proposed encoder. Smaller the DMOS means the difference of the perceptual quality between two video sequences is very small. Table 6.6 shows the averaged DMOSs of the fifteen viewers for all the test sequences.

From the experimental results, we can find following phenomena:

1. For *Cyclists*, *Crew*, *Sailormen*, *Tractor* and *Riverbed*, the DMOS gains are very tiny. It means the perceptual quality of two sequences is almost same. The reason is these sequences are very smooth, so JND value is small, resulting in lower perceptual redundancy. Thus, there is no obvious perceptual quality gain between JM and the proposed method.

2. For *City*, *Harbour*, *Shields*, *Stockholm* and *Calendar*, viewers can see the difference between two methods. Since DMOS scores are no more than 10, the perceptual quality gain is still not significant. The reason is that these sequences contain a lot of texture and have higher JND thresholds, resulting higher perceptual redundancy, but human eyes are usually not sensitive to the texture area. Thus, it is a little more difficult to see the perceptual quality gain.

6.3.3 Discussion

Since JND stands for the maximum distortion which cannot be perceived by the human eyes, transparent video coding is a good application of JND techniques. For the videos with different motion, texture and content, different perceptual redundancy can be removed efficiently according to the corresponding JND map. Thus, the bit-rate can be reduced with the same perceived quality.

JND based technique has its limitation. It is difficult to achieve higher perceptual quality at the fixed bit-rate. In order to achieve this task, JND based technique can be combined with other perceptual coding tools, such as ROI (region of interest) coding, perceptual rate control, etc., to achieve better performance.

6.4 Summary

In this chapter, a perceptually transparent image compression method is proposed based on the JND model. Not only statistical redundancy, but also perceptual redundancy existing in the images can be removed effectively. The experimental results show that the images compressed by the proposed method cannot be distinguished from the original ones. Moreover, compared with other lossless and near lossless algorithms, the proposed method can save significant amount of coding bits.

The proposed JND model is also applied in video coding. Based on the JND model, a perceptual video coding technique for H.264 is implemented. Benefited by the advantages of H.264 and HVS, the statistical and perceptual redundancy can be removed efficiently. Experimental results show that compared with the original H.264 encoder, the perceptually transparent encoder can reduce bit-rate significantly with the same perceived quality.

Part of the work in this chapter was published in *ISCAS2009*, entitled "The Perceptually Transparent Coding for Image", and was submitted to *IEEE Transactions on Circuits and System for Video Technology* as a brief paper, entitled "The Perceptually Transparent Coding for Image".

Chapter 7

Image Compression by Inverse and Forward Texture Syntheses

7.1 Introduction

Texture information, which prevails in image and video especially in the landscape scenes, are those objects (e.g., grass, water, cloud and tree) at far distance whose details are homogeneous [Guo et al., 2007]. When people view such textured image and video, they usually have a macro-view of the texture regions in terms of their shape, lighting and shading, whereas the micro-structures and the details of texture are often ignored and have little significance on human perception. Unfortunately, this perceptual property has been ignored in compression for long time. In traditional image and video compression schemes (e.g., the state-of-the-art JPEG 2000 and H.264/MPEG-4 AVC), texture regions are processed the same way as other regions, where compression performance is evaluated by distortion of all pixels versus the bits to code them. Therefore, compressing texture regions will cost many bits because their micro structures and details contain a great deal of high frequency information although they are insignificant in human perception. Is there any way to compress texture efficiently by taking visual perception into account?

Texture synthesis should be a promising approach to be utilized in texture compression. So far, many successful texture synthesis algorithms were presented, which provide an alternative way to generate a large size texture region from a given small texture sample and can keep the properties of synthesized texture consistent with the given sample. They can be classified into four categories. The pixel-based approaches synthesize a new texture pixel by pixel [Efros and Leung, 1999] [Wei and Levoy, 2000]. The patch-based techniques generate the texture tile by tile and are much faster than the pixel-based approaches [Liang et al., 2001] [Cohen et al., 2003]. The optimization-based methods synthesize the texture by optimizing an energy function [Kwatra et al., 2005], whilst those of the model-based approaches use parametric models to analyze and synthesize texture [Portilla and Simoncelli, 2000] [Heeger and Bergen, 1995].

Although there are many texture synthesis approaches, they are seldom applied

to handle texture regions of natural image and video in compression because it poses many new challenges to traditional texture synthesis. Firstly, in order to reconstruct texture by using texture synthesis, we should summarize the input texture as a small sample and/or as few parameters as possible. However, the existing texture synthesis methods either lack the analysis for texture, or cannot model texture concisely and accurately. Secondly, the texture of image and video is captured in a real environment and contains complicated shape, lighting and shading. Moreover, in a real environment, texture exhibits the strong scaling property dependent on their distances to camera. It is difficult to be described by the several ideal texture samples. The pioneering work done by Dumitras et al. incorporated the model-based texture synthesis into video compression [Dumitras and Haskell, 2003] [Dumitras and Haskell, 2004]. This method analyzes the selected texture region as some filter coefficients and the new texture will be synthesized at the decoder side relying on these coefficients. This method is only effective for the stochastic textures and is not good for the structural or periodic textures. The perceptual quality is also not so appealing.

In this chapter, we no longer assume that the texture regions of natural image and video can be generated from a set of pre-defined samples or simply modeled by some parameters. Taking the complication of textures in natural image and video into account, we propose to first analyze input texture and generate an adaptive small sample to represent the features of the input texture. Obviously, the sample generation is not a simple task like down-sampling that would result in considerably blur and quality degradation in the synthesized texture. As a matter of fact, it is a tough problem to generate a sample as small as possible, while preserving all the features of the input texture at high quality without resolution degradation. To achieve it, this paper adopts the new advance on texture syntheses, namely, inverse texture syntheses approach proposed by Wei et al in [Wei et al., 2008a] [Han and Wei, 2007]. The inverse texture synthesis operates in the opposite way with respect to the forward synthesis: given a large global variant texture, a small sample that best summarizes the original can be generated automatically.

After the sample is generated, we further propose to extract some auxiliary parameters by using a motion-estimation-like technique. These parameters describe the mapping from the input texture to the generated small sample. Both the sample and auxiliary data are compressed and transmitted to synthesize the input texture at the decoder side. The sample is much smaller than the original but contains enough detailed information to reconstruct the original. Although some additional bits are needed to send the auxiliary data, our proposed scheme can reduce the bit-rate greatly compared with JPEG at similar visual quality level. And at the same bit-rate, the perceptual quality of our method outperforms that of JPEG and JPEG2000.

Recently, remarkable progresses have been made in computer vision and graphics that give image compression a brilliant future to improve perceptual quality. Several non-traditional compression schemes have been presented in the literature [Rane et al., 2003; Liu et al., ; Xiong et al., 2007], which incorporate image inpainting into compression. They can efficiently represent smooth and structured regions in images and videos. But they do not work well for texture regions. Our proposed compression scheme is a good supplement to these methods and can incorporate with them to achieve a complete story on utilizing computer vision techniques for compression.

The rest of the chapter is organized as follows. Section 7.2 discusses texture model and the inverse texture synthesis technique. Section 7.3 shows how to extract the auxiliary information and map the original image to the sample. In Section 7.4, our proposed compression scheme as well as the implementation details is introduced. In Section 7.5, experimental results are shown. Finally, conclusion and the future work are presented in Section 7.6.

7.2 Texture Model

The goal of the image compression is to achieve better visual quality at the given bits budget, or to utilize fewer bits to reach equivalent visual quality. Thus, two problems are of concern to researchers. One is how to reduce the bit-rate. Another one is how to increase the quality of reconstructed images. When texture synthesis and image compression are jointly considered in an integrated coding system, two main problems should be addressed. The first one is how to use as few information as possible to represent the texture. This problem will affect how few the bits are used to compress the texture. The second one is how to make the condensed information represent the texture more accurately. This problem will guarantee how good the quality of the reconstructed images are. These two problems can be considered as a fundamental issue: how to model the texture exactly and efficiently. Moreover, our objective is that we can handle not only artificial synthetic textures, but also the natural textures. This makes our work much more challenging.

Portilla and Simoncelli [Portilla and Simoncelli, 2000] proposed a parametric model to represent the texture, which is an improvement of the model in [Heeger and Bergen, 1995] and is considered as the best analytical model so far. It is based on the first and second order properties of joint wavelet coefficients and provides impressive results. It can describe both stochastic and some repeated textures quite well, but usually fails to model some highly structured patterns. Unlike artificial synthetic textures, natural textures are very complex. They contain a lot of high frequency and highly structural information. It is difficult to model these patterns by only some limited parameters. Figure 7.1 shows the re-synthesis results by this method. We can see the artifacts are

Figure 7.1: *Synthesis results on complex structured textures by Portilla's method [Portilla and Simoncelli, 2000]*

very obvious and the quality is unacceptable for the image compression.

In many texture synthesis methods [Efros and Leung, 1999; Wei and Levoy, 2000; Liang et al., 2001; Cohen et al., 2003], texture is also modeled as a Markov Random Field (MRF). The probability distribution of brightness values for a pixel only depends on its neighborhood. The probability of the pixel x can be formulated as

$$P(x|x_1, x_2, \cdots, x_N) \qquad x_i \subseteq N(x) \tag{7.1}$$

where $N(x)$ describes the all the neighborhoods of x. The size of the neighborhood is a free parameter that specifies how stochastic the texture is. However, the analytical form of this texture model is not available. It is very difficult to be used in texture compression to achieve better results.

For some homogeneous or periodic textures, texton is a good description for the textures. Texton means texture element, which can be extracted by just cropping a sample from texture. Charalampidis [Charalampidis, 2006] proposed a texton extraction method by using signal processing techniques. The periodicity is determined from the image autocorrelation in frequency domain. This method is also effective for quasi-periodic textures. But unfortunately, in nature scene, this kind of texture is too special to appear. Thus, this disadvantage limits the application of this method in image compression field.

Natural textures usually vary dramatically to reflect the complicated real world. It is inhomogeneous in many aspects, including lighting, direction, density, linearity, frequency, phase, etc. Therefore, it is difficult to represent the texture with a single method. Down-sampling texture into a small sample maybe is a choice, but lots of high frequency components are lost during the down-sampling process and are very difficult

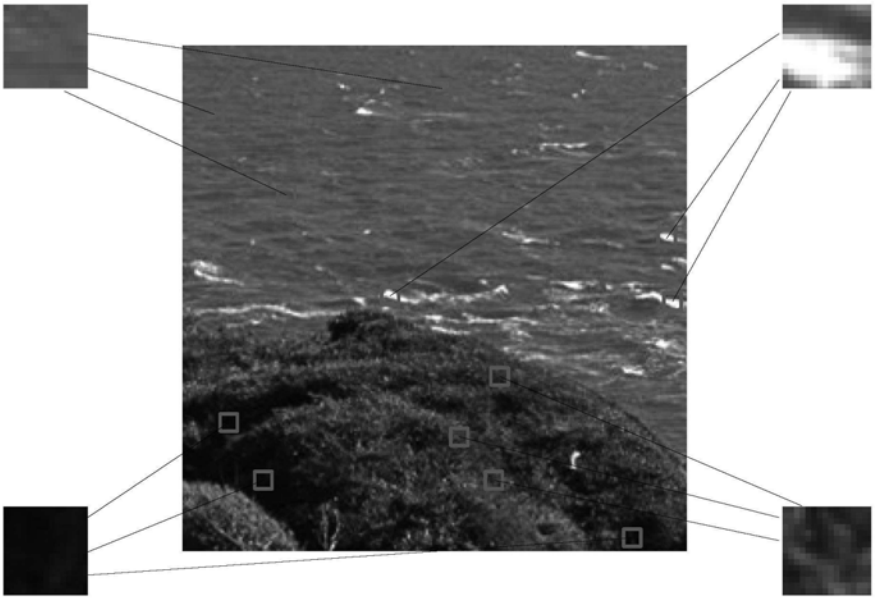

Figure 7.2: *Spatial redundancy of the texture*

to be recovered, thus blur artifacts are very easy to be introduced. High frequency information is very important for image, especially for texture, because human eyes are very sensitive for the blur error in the texture region. We wonder whether a model exists which can summarize all the necessary information in the texture, especially the high frequency details. In Figure 7.2, we can find that texture not only has local correlations, but also contains some globally repeated information. These similar patterns distribute far from each other. This kind of spatial redundancy is not easy to be handled by the traditional image models. One straightforward idea comes out: could we use a smaller compaction to model the total texture where every similar region in the original texture will be clustered into a small region in the compaction? The mathematical description for this model is shown in Eq. (7.2).

For a given texture X and a positive integer M

$$\exists D \subset R^2 \text{ with size } M \times M, \quad s.t. \sum_{\substack{N_{pi} \subset X \\ \hat{N}_{pi} \subset D}} ||N_{pi} - \hat{N}_{pi}||^2 \text{ minimum} \tag{7.2}$$

where, N_{pi} are any selected small neighborhoods in texture X, $\hat{N}_{pi}$ are the corresponding neighborhoods in the small region D. $||\cdot||$ is the L2 norm. Upper model means, for any given texture X, there is a condensed sample D which guarantees that for any selected neighborhood in X, the best matching neighborhood can be found in D to make the total distortion minimum. If we obtain this condensed texture, it is very easily to use very few bits to describe and reconstruct the original texture. Thus, how to generate this compaction is the key problem. Fortunately, inverse texture synthesis method [Wei

et al., 2008a; Han and Wei, 2007] gives a good solution for this model.

The inverse texture synthesis works in the opposite direction with respect to traditional forward texture synthesis. This process can be formulated as an optimization problem. Specifically, for a given original texture X, the objective is to compute a small sample Z with user-specified size by minimizing the following function:

$$\Phi(x;z) = \frac{1}{X^*}\sum_{p\in X^*}\left|x_p - z'_p\right|^2 + \frac{\alpha}{Z^*}\sum_{q\in Z^*}\left|x'_q - z_q\right|^2 \tag{7.3}$$

z and x stand for pixel values in Z and X, respectively. q and p run in the subset Z^* and X^* of Z and X. x_p and z_q are the neighborhoods around p and q. z'_p is the most similar neighborhood in Z with respect to x_p. x'_q is the most similar neighborhood in X with respect to z_q. α is a user-tunable weighting factor and works well for most textures by being set as 0.01. This energy function is solved by an expectation-maximization (EM) solver to get our expected sample.

As shown in Eq. (7.3), the function consists of two terms. Although they look very similar, they serve totally different purposes. The first term is called the inverse term, which measures the local similarity of a set of neighborhoods in X with respect to Z. This term is very important. Without this term, the resulting sample may lose some important information in the original texture. The second term is called the forward term. By optimizing this term, we can guarantee that for each selected neighborhood in Z, we can find a most similar patch in the original X to ensure that the total distortion energy is minimum. The reason for introducing this term is to let all the areas in the sample be utilized sufficiently. For example, if only the inverse term exists, it might happen that all the selected neighborhoods in the original are mapped to a small area in the sample, then causing other areas to be wasted. Due to the limitation of the chapter length, please refer [Han and Wei, 2007] for more details.

It is very straightforward that the energy function value varies according to the size of the sample. With the increase of sample size, the value of the energy function will be lower. Thus, more information in the original texture can be preserved in the sample. We have found that the quarter size of the original is a good choice for the sample and it works well for most textures we have tested.

7.3 Auxiliary Information Extraction and Processing

Although the sample is a good summarization of the original texture, for most general textures, it cannot re-synthesize the original directly from the sample. Researchers in computer graphics field usually utilize control map as the constraints to guide the re-synthesis process. For the homogenous texture, control map is a constant, but for most of general textures which are inhomogeneous and globally variant, control map

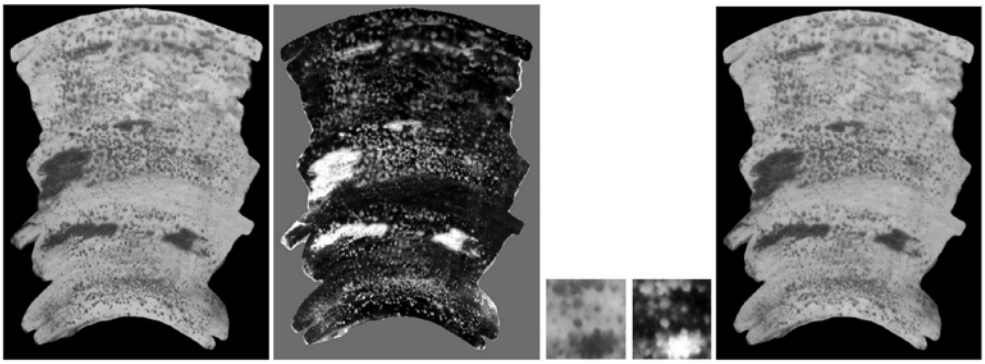

Figure 7.3: *Inverse texture synthesis and re-synthesis on image Banana. From left to right: the original (720×540), control map for the original, the compaction (64×64), control map for the compaction, the re-synthesized image.*

becomes a very complex one. It could be chrominance information, degree map [Wang et al., 2006], spatially-varying parameters [Gu et al., 2006], context information [Lu et al., 2007], and so on. It covers almost all the features used in image processing and usually is obtained by computer vision methods. So far, there is not a general way to extract the control maps for the different kinds of the textures. For many natural textures, control map is hard to generate. Moreover, control map generally is a large size image which contains much high frequency information as shown in Figure 7.3. It is not economic to use control map as the auxiliary information in image compression.

Because the inverse texture synthesis technique is based on the L2 norm, the resulting sample is an optimal solution that every neighborhood in original image can find the best match in the sample with the minimum distortion. Based on this observation, for each block in the original texture, we can use the best matching location in the sample as our auxiliary information. The block size will influence the coding performance significantly. If the block size is larger, fewer bits will be consumed to encode the location information, but matching error will be larger. If the block size if smaller, matching error will be smaller, but the location information will utilize more bits. By considering the tradeoff between rate and quality, the block size is chosen as 4×4 in our scheme. For each block, we can get the best matched position in the sample by using the block matching method. In order to enhance the quality, the full search is used here. The whole process is formulated as the following equation

$$(pos_x^*, pos_y^*) = arg\min(\sum_{i=1}^{4}\sum_{j=1}^{4}\left|O_{i,j} - C_{i+pos_x,j+pos_y}\right|^2) \tag{7.4}$$

$O_{i,j}$ is the intensity value at the i-th column and the j-th row in the original block. $C_{i+pos_x,j+pos_y}$ stands for the intensity value at the $(i + pos_x)$-th column and the $(j + pos_y)$-th row in the sample. (pos_x, pos_y) is the searching position and runs in

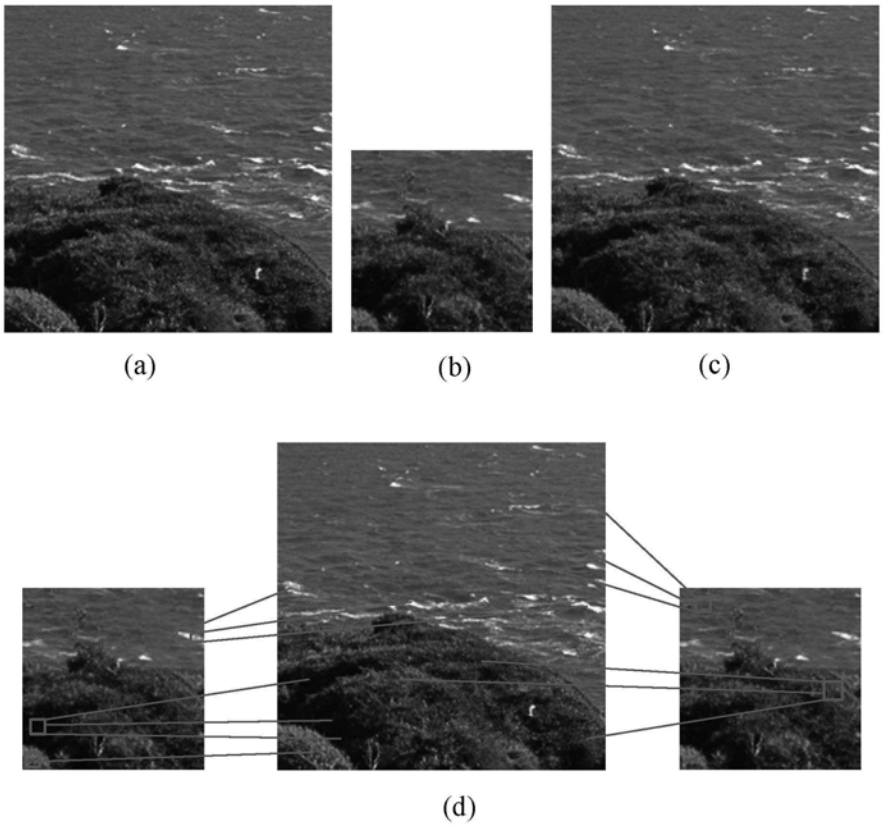

Figure 7.4: *Illustration of mapping: (a) luma of the image Water01; (b) extracted sample; (c) reconstructed luma; (d) a part of the mapping between the sample and the input image*

all the areas of the sample. (pos_x^*, pos_y^*) is the best matching position which makes the matching error minimum.

Figure 7.4 shows an example of our compression scheme. Input image is condensed into a sample which keeps enough detailed information in the original texture. During a block mapping process, every block in the input image can find its corresponding position in the sample. Auxiliary information records the x and y coordinates of all the best matched positions. We use fixed length coding to encode the auxiliary information because the variable length coding is not economic for our case. The location information retrieval method in our algorithm is not like the traditional motion estimation. Firstly, searching range is large. Secondly, correlation among location information is not high. For example, for a 64×64 texture, the sample size is 32×32. 5 bits are needed for fixed length coding (FLC). If we use variable length coding (VLC), Figure 7.5 shows the histograms of difference signals of x and y coordinates, where the entropies are is 5.13 and 3.97, respectively. One can observe that variable length coding cannot provide any benefit in our scheme. If considering the overhead of the code table, even the lengths of variable codes are longer than that of the fixed length coding. So we use fixed length coding in our scheme.

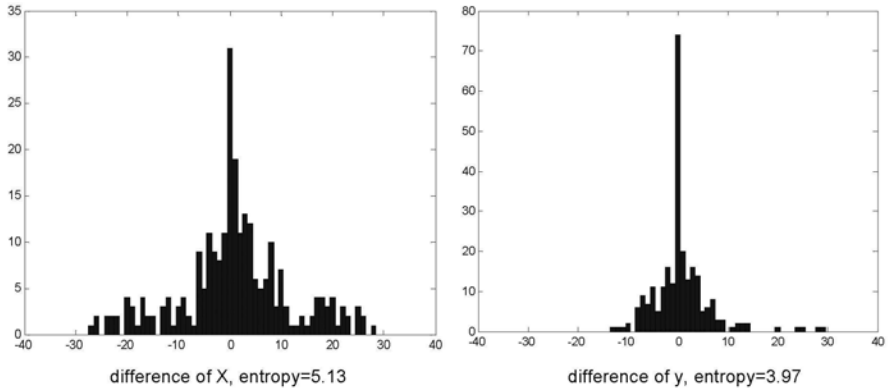

Figure 7.5*: Histogram for the difference of x and y coordinates.*

7.4 Proposed Texture Compression Scheme

With the above techniques, the whole structure of our proposed compression scheme is depicted in Figure 7.6. The input image is first decomposed into luminance and chrominance components. The chroma component is handled in the traditional way. The luma is divided into equal size sub-regions. Each sub-region is processed by the inverse texture synthesis module to generate the small sample. Auxiliary information is also extracted for each sub-region. Then all the samples are combined into a big sample according to their distribution. Finally, auxiliary information, sample and chroma component are encoded into the bit-stream. At the decoder side, the forward texture synthesis is performed successfully to restore the original image, relying on the auxiliary information and the received sample.

There are two reasons why the image is separated into some sub-regions. The first one is that the smaller region can be considered approximately homogenous and the inverse texture synthesis can achieve better optimization result. Another reason is that the smaller the region, the smaller the corresponding sample, thus the fewer bits are consumed for encoding the auxiliary information. However, sub-region should not be too small since no much redundancy can be exploited in a small region. Empirically, the size of the sub-region is no smaller than 128×128.

Here, three kinds of information need to be encoded: chrominance, auxiliary information and the integrated sample. Auxiliary information uses fixed length coding as explained in the previous section. The integrated sample and the chrominance component are encoded by the traditional codec which is JPEG in our implementation. The reason why we treat chrominance different with luminance is: chrominance usually contains low high-frequency energy and looks smooth, but luminance has much higher high-frequency energy. If these two components are optimized together by the inverse texture synthesis, the quality of the chrominance will be affected more seriously because, under the same distortion level, the smoother texture will be more sensitive

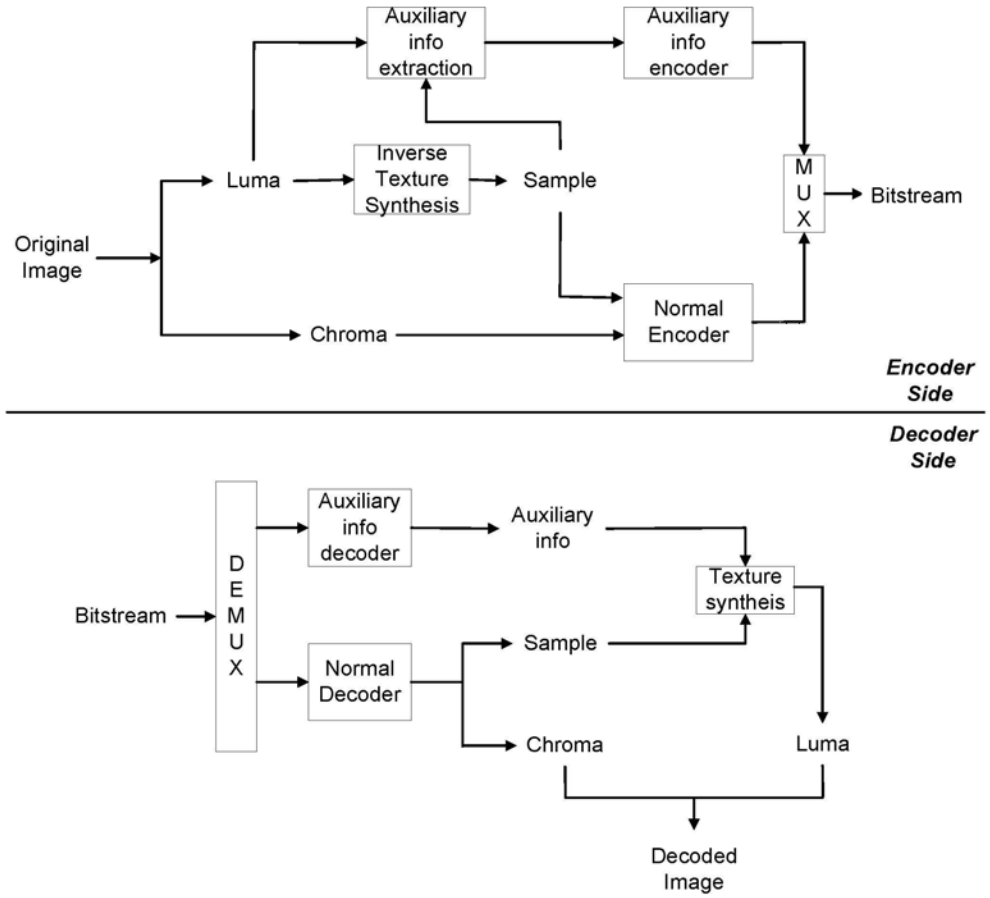

Figure 7.6: *The framework of proposed compression scheme.*

for the error. Moreover, generally chrominance consumes fewer bits than luminance, so we encode it by the conventional way.

At the decoder side, the luminance is easy to be reconstructed by copying the corresponding block from the received sample according to the auxiliary information. Then, the original image is recovered depending on the reconstructed luminance and the decoded chrominance. Our re-synthesis method can be considered as a special case of the patch-based texture synthesis where the location information is utilized to control the synthesis process. Since the human eye usually is not sensitive to the distortion in the texture region and the 4×4 patch is small enough, our method does not introduce any blocky artifacts. Moreover, our decoder is very simple and the computation load is very low compared to the inpainting based image compression methods [Rane et al., 2003; Liu et al., ; Xiong et al., 2007].

Subjective score	Description
-3	The right one is much worse than the left one
-2	The right one is worse than the left one
-1	The right one is slightly worse than the left one
0	The right one has same quality as the left one
1	The right one is slightly better than the left one
2	The right one is better than the left one
3	The right one is much better than the left one

Table 7.1*: Comparison scale for subjective quality evaluation*

	JPEG (bpp)	Ours (bpp)				Saving ratio	Subjective score
		Chroma	Sample	Assis. info	Total		
Water01	2.09	0.13	0.48	0.75	1.36	-35%	-0.30
Water02	2.29	0.08	0.57	0.75	1.40	-39%	-0.10
Grass01	2.83	0.17	0.61	0.75	1.53	-46%	0.10
Grass02	2.37	0.17	0.51	0.75	1.43	-40%	0.00
Grass03	3.75	0.21	0.70	0.75	1.66	-56%	-0.30
Average	2.67	0.15	0.57	0.75	1.48	-43%	-0.12

Table 7.2*: Bit rate saving ratio of our scheme compared with JPEG*

7.5 Experimental Results

The experiments are conducted to evaluate the performance of the proposed natural texture image compression in terms of bit-rate and visual quality. Five color images were used in our experiments: two are from the Vistex database [Vistex, 2002] with the size 256×256 and three are natural images taken by us with the size 512×512. The traditional image codec (JPEG and JPEG 2000) are used as the references for comparisons.

Since our scheme uses JPEG as the traditional codec, the first experiment compares the generated bit rates of our scheme and JPEG with the quantization parameter (QP) of JPEG set to 75, which is the default value in JPEG and can achieve almost perceptually lossless quality. Bit-rate saving ratios are listed in Table 7.2. One can observe that our method can save up to 56% bit-rate compared with JPEG. Since the actual sample size is about 1/4 of the input texture image, the actual rate for the sample coding in our scheme is also about 1/4 of the JPEG rate. The rate for the auxiliary information coding is 0.75 bpp in our scheme. Figure 7.7 shows the perceptual quality of our scheme compared with that of JPEG. One can observe that they have almost same perceptual quality.

For a more convincing evaluation, the subjective viewing tests were conducted based on "Adjectival categorical judgement methods" recommended by ITU-R BT.500-11 standard [ITU-R, 2002]. In each test, two images were displayed on the screen side

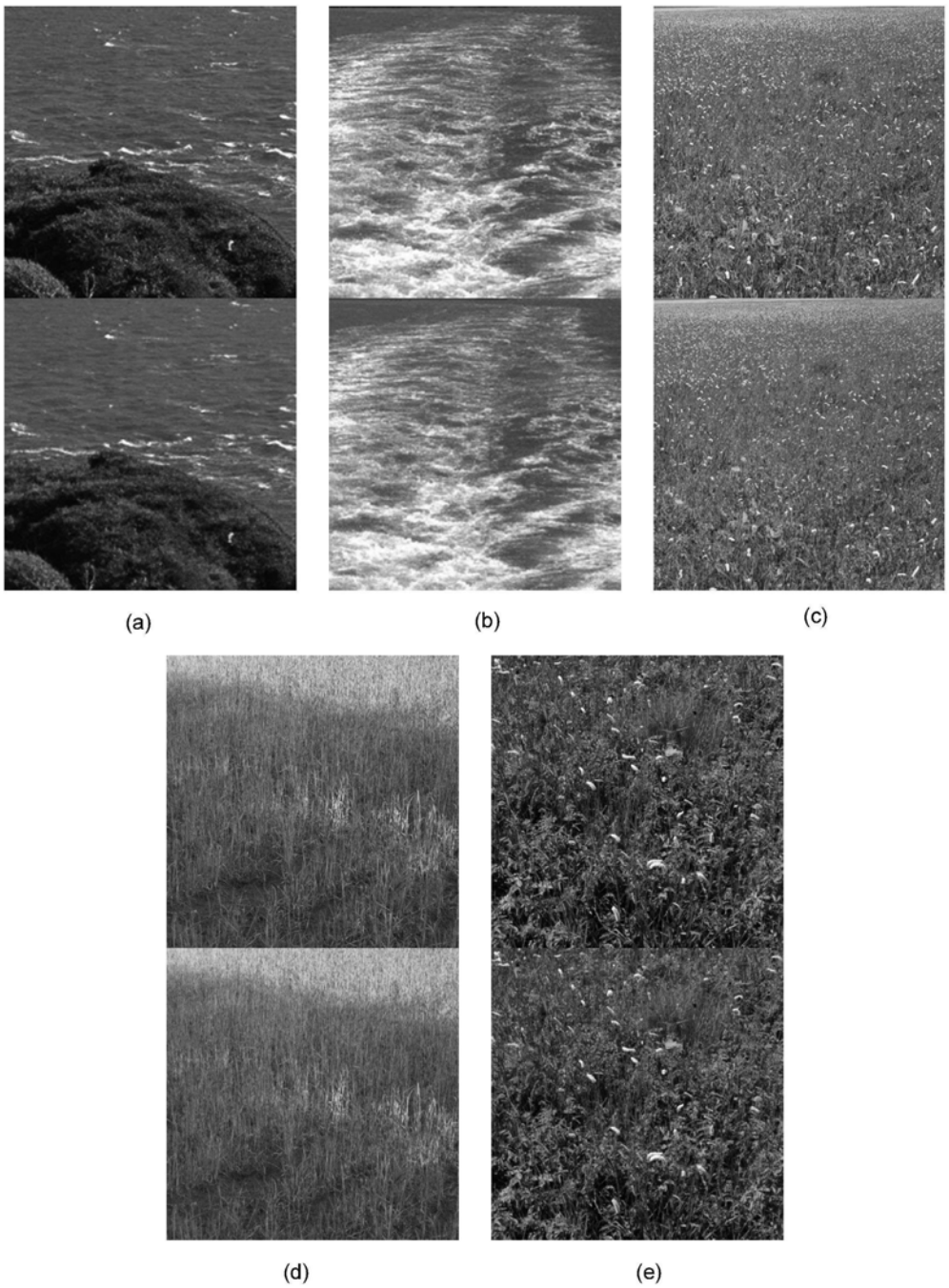

Figure 7.7: *Comparisons with JPEG: (a) Water01(256×256); (b) Water02(256×256); (c) Grass01(512×512); (d) Grass02(512×512); (e) Grass03(512×512). The top is the reconstructed image by JPEG and the bottom shows the reconstructed image by our scheme.*

by side (left is the JPEG coded image and right is the image coded by the proposed method). Ten subjects were asked to give quantitative scores for all the image pairs, using the continuous quality comparison scale shown in Table 7.1. The subjective scores are shown in Table 7.2 and the average score is only -0.12. It means the images coded by these two methods have very close subjective quality.

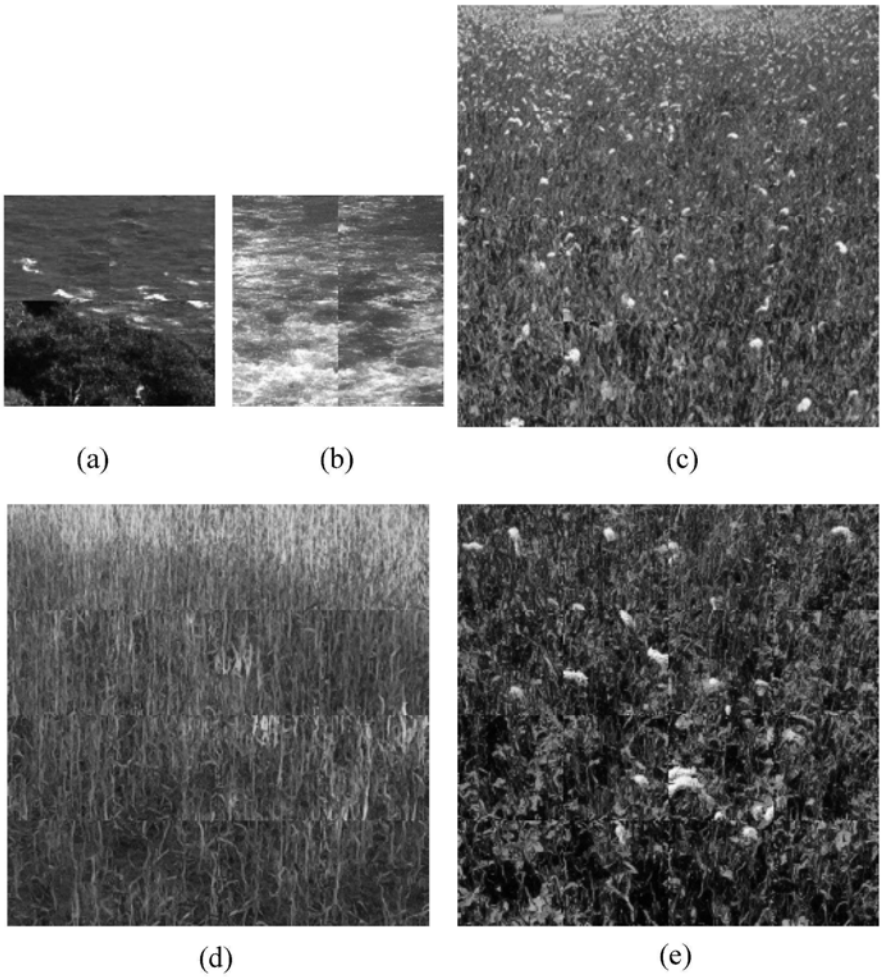

Figure 7.8: *Small texture samples in our proposed scheme. (a) Water01(128×128); (b) Water02(128×128); (c) Grass01(256×256); (d) Grass02(256×256); (e) Grass03(256×256).*

Figure 7.8 show the example samples in the first experiment. One can observe that the samples contain almost all information of the input originals although the size is only 1/4 of the original size.

In the second experiment, we compare the perceptual quality of our scheme with JPEG and JPEG2000 at the same bit-rate. More results are shown in Figure 7.9. At the same bit-rate, the perceptual quality of our method looks better than those of JPEG and JPEG2000. Blur artifacts can easily be observed in the image compressed by JPEG and JPEG2000, which are the most annoying artifacts for texture region. Our method retains the details much better at the same bit-rate. Since the texture region contains a lot of high frequency energy which is very hard to compress, in the lower bit-rate case, these traditional codecs will destroy high frequency components due to quantization error, resulting in blur artifacts. Our method is based on inverse texture synthesis which makes the similar texture regions in the original clustered together in the sample, so the detailed information is preserved and better perceptual quality can

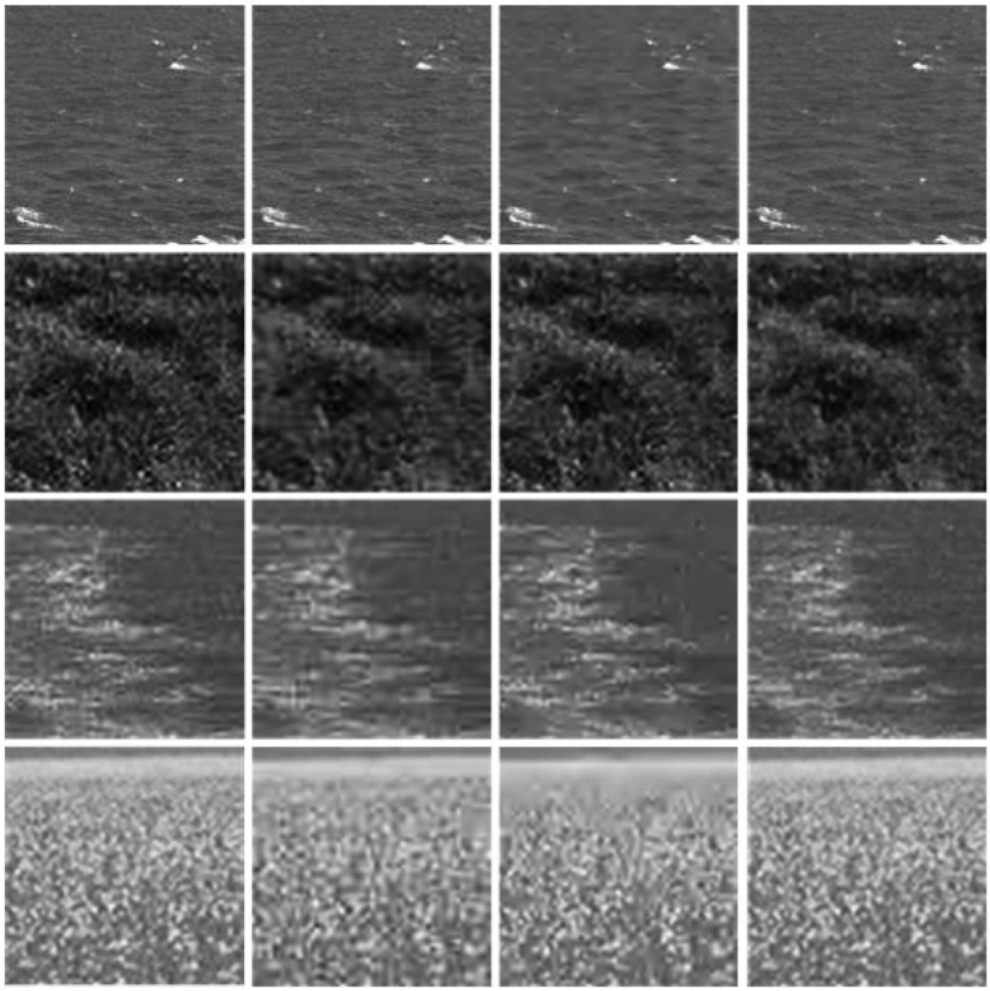

Figure 7.9: *Comparisons of details. From left to right: Original; JPEG; JPEG2000; Ours. The latter three images are at the same bit rate.*

be achieved.

7.6 Summary

In this chapter, a novel image compression scheme is proposed which is based on the inverse texture synthesis and exploits the visual redundancy in the texture region more effectively. The experimental results show that our proposed scheme can reduce bit-rate greatly compared to JPEG at similar visual quality. At the same bit-rate, the perceptual quality of our method is better than those of JPEG and JPEG2000. Further improvement of our scheme is possible. Firstly, our scheme can handle texture region efficiently, where previous computer vision based compression schemes usually do not work well. Our scheme could be incorporated into other inpainting based methods [Rane et al., 2003; Liu et al., ; Xiong et al., 2007] to achieve better performance. Secondly, our algorithm is very flexible and adaptable. It can be easily implemented in

other traditional codec, such as H.264 intra coding, to improve the coding performance further.

All of the work in this chapter was submitted to *ICIP2009*, entitled "Image Compression by Inverse and Forward Texture Syntheses ", and was submitted to *IEEE Transactions on Circuits and System for Video Technology* as a regular paper, entitled "Image Compression by Inverse and Forward Texture Syntheses".

Chapter 8

Conclusions

With the development in the past two decades, the traditional image and video compression are meeting the bottleneck. Two fundamental problems exist in current compression schemes: first, although more and more complex techniques are introduced in compression standards to increase the compression efficiency, the improvement is very limited. Secondly, only the statistical redundancy among pixels is considered as the objective of the optimization, and the perceptual redundancy is almost neglected. In order to solve above problems, several efficient and perceptual picture coding techniques have been introduced and discussed in the thesis. The contributions of the research work in the thesis are summarized here and we also discuss several directions for the future work.

8.1 Contributions of the Thesis

A fast intra mode selection algorithm based on the edge classification and rate-distortion estimation [Wei et al., 2008b; Wei et al., 2007a; Wei and Ngan, 2007; Li et al., 2008] is presented in Chapter 2. By using a fast edge detection method based on non-normalized Haar transform (NHT), edges for each sub-block can be extracted. By using local edge information, only a few intra modes are chosen as mode candidates. A fast RDO algorithm is also proposed in this chapter based on accurate rate-distortion estimation model and the fast intra mode RDO method. By combining these two methods, computational load is reduced remarkably. Experimental results show that the fast intra mode selection scheme can shorten the encoding time significantly without much loss of bit-rate and visual quality.

In the techniques adopted in H.264, multiple block size mode decision in the inter frame coding is an important part and could achieve nearly 20% bit rate reduction. However, searching all block sizes greatly increases the computational complexity of the H.264 codec. Therefore, an efficient mode selection method is necessary. In Chapter 3, we propose a fast inter mode decision algorithm [Wei and Ngan, 2006]. The main techniques include: Pskip mode early detection based on transform domain, mode prediction, and early termination using adaptive threshold. We also use a post-search

technique to keep coding quality and intra mode skip detection technique to save computational time. Compared with full mode selection and other fast algorithms, our new algorithm maintains the image quality quite well with almost the same bit rate.

To build a high efficient multimedia processing platform by combining H.264 and an efficient embedded processor is significant in engineering and has a lot of market values. The implementation and optimization of H.264 baseline profile on mobile device [Wei et al., 2007b] is discussed in Chapter 4. Our work is to implement a real time H.264 baseline encoder and decoder on a HP 4700 PDA which has a powerful embedded processor PXA27x produced by Intel. By the optimization at three levels, we have successfully implemented the H.264 baseline codec on mobile device and made great improvement in terms of coding speed with very little performance degradation in terms of PSNR and bit-rate. Experimental results show that the speed of our codec increases significantly after optimization (much more than 25 frames per second). In the QCIF (176×144) resolution, our codec can run in real time on the mobile device. Based on this real time H.264 codec, a mobile video conferencing system is implemented. Our system works in the duplex mode. Since the implementation is software based, the system can be easily transplanted on other mobile devices, such as mobile phone, laptop, etc. It can also be applied in other wired or wireless networks, such as 3G or 4G communication systems.

In image and video processing field, an effective compression algorithm should remove not only the statistical redundant information but also the perceptually insignificant component from the pictures. Just-noticeable distortion (JND) profile is an efficient model to represent those perceptual redundancies. Human eyes are usually not sensitive to the distortion below the JND threshold. In Chapter 5, a DCT based JND model for monochrome pictures is proposed [Wei and Ngan, 2008b; Wei and Ngan, 2008c; Wei and Ngan, 2008d]. This model incorporates the spatial contrast sensitivity function (CSF), the luminance adaptation effect, and the contrast masking effect based on block classification. Gamma correction is also considered to compensate the original luminance adaptation effect which gives more accurate results. In order to extend the proposed JND profile to the video images, the temporal modulation factor is included by incorporating the temporal CSF and the eye movement compensation. Furthermore, a psychophysical experiment is designed to parameterize the proposed model. Experimental results show that the proposed model is consistent with the human visual system (HVS). Compared with the other JND profiles, the new model can tolerate more distortion and has much better perceptual quality. This model can be easily applied in many related areas, such as compression, watermarking, error protection, perceptual distortion metric, and so on.

In Chapter 6, based on the proposed spatial JND model, we present a perceptually

transparent image compression method for monochromatic images [Wei and Ngan, 2008a; Wei and Ngan, 2009], where the quantization factor can be tuned for each block according to the JND threshold. This makes the quantization error lower than the JND threshold. The proposed algorithm is also extended to the color images. The experimental results show that the images compressed by the proposed method are hardly distinguished from the original images. Therefore, the proposed method can achieve perceptually transparent quality for images. Moreover, the bit-rate of the proposed algorithm is less than that of many state-of-the-art lossless and near-lossless codecs. Moreover, based on the proposed spatio-temporal JND model, a transparent video codec is implemented in H.264. Compared with the original H.264 encoder, the perceptually transparent encoder can reduce bit-rate significantly with the same visual quality.

In picture compression, texture regions usually consume many bits because of the rich high frequency information. Instead of directly compressing the texture regions pixel by pixel, texture synthesis provides an alternative but more promising way. However, traditional texture synthesis approaches which take one sample from a set of pre-defined textures to generate a texture picture are not suitable for representing textures in natural images, because they usually vary broadly in terms of view angle and lighting even for the same scene. To solve this problem, a image compression scheme based on the inverse and forward texture synthesis [Wei et al., 2008c; Wei et al., 2009] is presented in Chapter 7. In the proposed scheme, input texture is first analyzed to generate a small sample. Although the sample size is small, it includes almost all features of the input texture without resolution degradation. At the same time, some auxiliary parameters are generated to describe the mapping from the small sample to the input texture. Both of them are compressed and transmitted to reconstruct input texture at the decoder. Experimental results show that our proposed compression scheme can achieve up to 56% bit-saving compared with JPEG at the similar perceptual visual quality. And at the same bit-rate, the perceptual quality of our method outperforms those of JPEG and JPEG2000.

8.2 Future Work

Some suggestions for future work are listed below:

- JND model for color pictures: The proposed JND model in Chapter 5 is only for the grey scale image/video or luminance component of the color image/video. The chrominance information is not considered in our proposed model. Few studies have been found in literatures to derive a complete JND model for color picture application, although color pictures attract the interests of human observers more than grey ones. An accurate color JND model should significantly facilitate the

perceptual quality measure and coding for color pictures, so it is very important to extend our proposed JND model to color pictures by taking into account CSF and other vision effects for color channels.

- Perceptual picture quality metric: The traditional picture quality metrics, such as mean square error (MSE) and peak signal-to-noise ratio (PSNR), have been widely criticized for not correlating well with the perceived quality measurement. For better perceptual error evaluation, a perceptual picture quality metric can be designed based on our proposed JND model, since JND can model HVS much better than those signal-processing based pixel-wise distortion metrics.

- Consideration of attention model: Normally JND is generated by using some low level and natural information of the image/video. However, human vision is very complex and related to the high level information. People usually pay more attention to some regions of interest, such as human faces, region with motion, region with high contrast etc. Thus, the normal JND map can be modulated by a human attention model to simulate HVS more accurately. The lower JND threshold can be given to the regions where people are interested in and higher JND threshold can be given to those uninterested regions.

- Computer-vision based picture compression scheme: In Chapter 7.1, a texture synthesis based image compression method is proposed, which is very effective for the texture compression. Several non-traditional compression schemes have been presented in literatures [Rane et al., 2003; Liu et al., ; Xiong et al., 2007], which incorporate image inpainting to compression. They can efficiently represent the smooth and structured regions in images and videos. But they do not work well for the texture regions. Our proposed compression scheme is a good supplement to these methods and can be incorporated to achieve a better solution on utilizing computer vision techniques for compression.

Bibliography

[Ahmad et al., 2004] Ahmad, A., Khan, N., Masud, S., and Maud, M. (2004). Efficient block size selection in h.264 video coding standard. *Electronic Letter*, 40(1).

[Ahumada and Peterson, 1992] Ahumada, A. and Peterson, H. (1992). Luminance-model-based dct quantization for color image compression. *Human Vision, Visual Processing, and Digital Display III, Proc. SPIE* , 1666:365–374.

[Bjontegaard, 2001] Bjontegaard, G. (2001). Calculation of average PSNR differences between RD-curves. *13th VCEG-M33 Meeting.*

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698.

[Charalampidis, 2006] Charalampidis, D. (2006). Texture synthesis: Textons revisited. *IEEE Trans. Image Processing*, 15:777–787.

[Chen and He, 2004] Chen, Q. and He, Y. (2004). A fast bits estimation method for rate-distortion optimization in H.264/AVC. In *Proc. Picture Coding Symp. (PCS).*

[Chen et al., 2002] Chen, Z., Zhou, P., and He, Y. (2002). Fast integer pel and fractional pel motion estimation for JVT, Joint Video Team(JVT) Docs. JVT-F017.

[Cheng and Chang, 2005] Cheng, C.-C. and Chang, T.-S. (2005). Fast three step intra prediction algorithm for 4×4 blocks in H.264. In *Proc. of IEEE Int. Symposium on Circuits and Systems (ISCAS).*

[Chiang and Zhang, 1997] Chiang, T. and Zhang, Y.-Q. (1997). A new rate control scheme using quadratic rate distortion model. *IEEE Transactions on Circuits and Systems for Video technology*, 7:246–250.

[Chin and Berger, 1999] Chin, Y.-J. and Berger, T. (1999). A software-only videocodec using pixelwise conditional differentialreplenishment and perceptual enhancements. *IEEE Transactions on Circuits and Systems for Video Technology* , 9:438–450.

[Chou and Chen, 1996] Chou, C.-H. and Chen, C.-W. (1996). A perceptually optimized 3-D subband codec for video communication over wireless channels. *IEEE Transactions on Circuits and Systems for Video Technology* , 6:143–156.

[Chou and Li, 1995] Chou, C.-H. and Li, Y.-C. (1995). A perceptual tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Transactions on Circuits and Systems for Video Technology* , 5:467–476.

[Cohen et al., 2003] Cohen, M. F., Shade, J., Hiller, S., and Deussen, O. (2003). Wang tiles for image and texture generation. In *proceedings of ACM SIGGRAPH 2003*, pages 287–294.

[Cornsweet, 1970] Cornsweet, T. (1970). Visual perception. *New York: Academic Press.*

[Daly, 1998] Daly, S. (1998). Engineering observations from spatiovelocity and spatiotemporal visual models . *Proc. SPIE* , 3299:180–191.

[Daly, 1992] Daly, S. J. (1992). Visible differences predictor: an algorithm for the assessment of image fidelity. *Proceedings of SPIE, Human Vision, Visual Processing, and Digital Display III*, 1666:2–15.

[Dumitras and Haskell, 2003] Dumitras, A. and Haskell, B. (2003). A texture replacement method at the encoder for bit-rate reduction of compressed video. *IEEE Trans. Circuits Syst. Video Technol.*, 13:163–175.

[Dumitras and Haskell, 2004] Dumitras, A. and Haskell, B. (2004). An encoder-decoder texture replacement method with application to content-based movie coding. *IEEE Trans. Circuits Syst. Video Technol.*, 14:825–840.

[Efros and Leung, 1999] Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 1033–1038.

[Foley and Boynton, 1994] Foley, J. M. and Boynton, G. M. (1994). New model of human luminance pattern vision mechanisms: analysis of the effects of pattern orientation, spatial phase and temporal frequency. *Computational Vision Based on Neurobiology,Proc. SPIE* , 2054:32–42.

[Girod, 1993] Girod, B. (1993). What's wrong with mean-squared error. In A.B.Watson, editor, *Digital Images and Hman Vison*, pages 207–220.

[Gu et al., 2006] Gu, J., Tu, C., Ramamoorthi, R., and et al. (2006). Time-varying surface appearance: acquisition, modeling and rendering. *ACM Trans. Graph.*, 25:762–771.

[Guo et al., 2007] Guo, C., Zhu, S. C., and Wu, Y. N. (2007). Primal sketch: integrating texture and structure. *Computer Vision and Image Understanding*, 106:5–19.

[Hahn and Mathews, 1998] Hahn, P. J. and Mathews, V. J. (1998). An analytical model of the perceptual threshold function for multichannel image compression. *IEEE International Conference on Image Processing (ICIP'98)*, 3:404–408.

[Han and Wei, 2007] Han, J. and Wei, L.-Y. (2007). Inverse texture synthesis. technical report in MSRA.

[He et al., 2001] He, Z., Kim, Y. K., and Mitra, S. (2001). Low-delay rate control for dct video coding via ρ-domain source modeling. *IEEE Transactions on Circuits and Systems for Video technology*, 11:928–940.

[He et al., 2002] He, Z., Kim, Y. K., and Mitra, S. (2002). Optimum bit allocation and accurate rate control for video coding via ρ-domain source modeling. *IEEE Transactions on Circuits and Systems for Video technology*, 12:840–849.

[He and Mitra, 2002] He, Z. and Mitra, S. (2002). A linear source model and a unified rate control algorithm for DCT video coding. *IEEE Transactions on Circuits and Systems for Video technology*, 12:970–982.

[Heeger and Bergen, 1995] Heeger, D. and Bergen, J. (1995). Pyramid-based texture analysis/synthesis. In *Proc.ACM SIGGRAPH '95.*

[HHI, 2005] HHI (2005). JVT model JM10.1. downloaded from http://iphome.hhi.de/suehring /tml/download/old_jm/.

[Hontsch and Karam, 2002] Hontsch, I. and Karam, L. (2002). Adaptive image coding with perceptual distortion control. *IEEE Transactions on Image Processing*, 11(3):213–222.

[Hontsch and Karam, 2000] Hontsch, I. and Karam, L. J. (2000). Locally adaptive perceptual image coding. *IEEE Trans. on Image Processing*, 9(9):1472–1483.

[Intel, 1999] Intel (1999). Intel Pentium 4 processor optimization reference manual. Intel Corp.

[Intel, 2001] Intel (2001). Next generation intel processor: Software developers guide. Intel Corp.

[Intel, 2002] Intel (2002). Intel wireless MMX^{TM} technology developer guide. Intel Corp.

[ISO/IEC, 1992] ISO/IEC (1992). Coding of moving pictures and associated audio - for digital storage media at up to about 1.5Mbit/s, ISO/IEC 11172-2.

[ISO/IEC, 1999] ISO/IEC (1999). Information technology-generic coding of audio-visual objects part2: visual, ISO/IEC 14496-2 (MPEG-4 Video).

[ISO/IEC, 2004a] ISO/IEC (2004a). Information technology – JPEG 2000 image coding system - Part 1: Core coding system, ISO/IEC 15444-1.

[ISO/IEC, 2004b] ISO/IEC (2004b). Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification (ISO/IEC 15948:2004).

[ITU-R, 2002] ITU-R (2002). Methodology for the subjective assessment of the quality of television pictures. *ITU-R BT.500-11.*

[ITU-T, 1994] ITU-T (1994). ITU-T recommendation H.261: Video codec for audiovisual services at p×64kbit/s.

[ITU-T, 1995] ITU-T (1995). Information technology - generic coding of moving pictures and associated audio information: Video, ITU-T recommendation H.262 and ISO/IEC 13818-2.

[ITU-T, 2001] ITU-T (2001). ITU-T recommendation H.263: Video coding for low bit rate communication.

[ITU-T and ISO/IEC, 1993a] ITU-T and ISO/IEC (1993a). Information technology – Coded representation of picture and audio information – Progressive bi-level image compression, ITU-T T.82, ISO/IEC 11544.

[ITU-T and ISO/IEC, 1993b] ITU-T and ISO/IEC (1993b). Information technology – Digital compression and coding of continuous-tone still images, ITU-T T.81 and ISO/IEC IS 10918-1.

[ITU-T and ISO/IEC, 2001] ITU-T and ISO/IEC (2001). Information technology – Lossy/lossless coding of bi-level images, ITU-T T.88, ISO/IEC 14492.

[ITU-T and ISO/IEC, 2003] ITU-T and ISO/IEC (2003). IInformation technology – Lossless and near-lossless compression of continuous-tone still images: Extensions, ITU-T T.87, ISO/IEC 14495.

[ITU-T and ISO/IEC, 2005] ITU-T and ISO/IEC (2005). Advanced video coding for generic audiovisual services, ITU-T recommendation H.264 and iso/iec 14496-10 avc: Version3.

[Jeon, 2003] Jeon, B. (2003). Fast mode decision to JVT, Joint Video Team(JVT) Docs. JVT-J033.

[Jia et al., 2006] Jia, Y., Lin, W., and Kassim, A. (2006). Estimating just-noticeable distortion for video. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(7):820–829.

[Kelly, 1979] Kelly, D. (1979). Motion and vision. II. Stabilized spatio-temporal threshold surface. *J Opt Soc Am* , 69:1340–1349.

[Khan et al., 2004] Khan, N., Masud, S., Ahmad, A., and Maud, M. (2004). Efficient scheme for motion estimation and block size mode selection in H.264. In *ISCIT'04*.

[Kim et al., 2006] Kim, C., Shih, H.-H., and Kuo, C.-C. J. (2006). Fast H.264 Intra-Prediction Mode Selection Using Joint Spatial and Transform Domain Features. *Journal of Visual Communication and Image Representation*, 17:291–310.

[Kwatra et al., 2005] Kwatra, V., Essa, I., and Bobick, A. (2005). Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24:795–802.

[Legge, 1981] Legge, G. E. (1981). A power law for contrast discrimination. *Vision Research*, 21:457–467.

[Legge and Foley, 1980] Legge, G. E. and Foley, J. M. (1980). Contrast masking in human vision. *Journal of the Optical Society of America*, 70:1458–1471.

[Li et al., 2008] Li, H., King N, N., and Wei, Z. (2008). Fast and efficient method for block edge classification and its application in h.264/avc video coding. *IEEE Trans. Circuits Syst. Video Technol.*, 18(6):756 – 768.

[Li and Ngan, 2006] Li, H. and Ngan, K. N. (2006). Fast and efficient method for block edge classification. In *Proc. ACM IWCMC2006 Multimedia over Wireless.*

[Liang et al., 2001] Liang, L., Liu, C., Xu, Y., Guo, B., and Shum, H.-Y. (2001). Real-time texture synthesis using patch-based sampling. *ACM Transactions on Graphcis.*

[Lin et al., 2003] Lin, W., Li, D., and Xue, P. (2003). Discriminative analysis of pixel difference towards picture quality prediction . *Proc. IEEE Int. Conf., Image Process.,* , 3:193–196.

[Liu et al.,] Liu, D., Sun, X., Wu, F., and et al. Image compression with edge-based inpainting. accpeted by IEEE Trans. Circuits Syst. Video Technol.

[Lu et al., 2007] Lu, J., Geogrhiades, A., Glaser, A., and et al. (2007). Context aware texture. *ACM Trans. Graph.*, 26.

[Lu et al., 2005] Lu, Z., Lin, W., Yang, X., Ong, E., and Yao, S. (2005). Modeling visual attention's modulatory aftereffects on visual sensitivity and quality evaluation. *IEEE Transactions on Image Processing*, 14:1928 – 1942.

[Lubin, 1995] Lubin, J. (1995). A visual discrimination model for imaging system design and evaluation. In Peli, E., editor, *Vision Models for Target Detection and Recognition*, pages 245–283. World Scientific Publishing Co. Pte. Ltd.

[Meng and Au, 2003] Meng, B. and Au, O. C. (2003). Efficient intra-prediction mode selection for 4x4 blocks in H.264. In *Proc. of IEEE Int. Conf. Multimedia and Expo. (ICME).*

[Nettravali and Haskell, 1988] Nettravali, A. N. and Haskell, B. G. (1988). Digital pictures : Representation and compression. *Plenum Press.*

[Ngan et al., 1989] Ngan, K. N., Leong, K. S., and Singh, H. (1989). Adaptive cosine transform coding of images in perceptual domain. *IEEE Transactions on Acoustics, Speech, and Signal Processing* , 37:1743–1750.

[Nill, 1985] Nill, N. B. (1985). A visual model weighted cosine transform for image compression and quality assessment. *IEEE Transactions on Communications*, 33(3):551 – 557.

[Ong et al., 2005] Ong, E., Lin, W., Lu, Z., and Yao, S. (2005). Colour perceptual video quality metric. *Proc. IEEE Int. Conf., Image Process.,*, pages 1172 – 1175.

[Painter and Spanias, 2000] Painter, T. and Spanias, A. (2000). Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451 – 515.

[Pan and Lin, 2004] Pan, F. and Lin, X. (2004). Fast intra mode decision algorithm for H.264/AVC video coding. In *Proc. IEEE Int. Conf. Image Processing (ICIP).*

[Pan and Lin, 2005] Pan, F. and Lin, X. (2005). Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding. *IEEE Transactions on Circuits and Systems for Video technology*, 15:813–822.

[Peterson et al., 1993a] Peterson, H. A., Ahumada, A. J., and B.Watson, A. (1993a). Improved detection model for DCT coefficient quantization. *Human Vision, Visual Processing, and Digital Display IV, Proc. SPIE* , 1913:191–201.

[Peterson et al., 1993b] Peterson, H. A., Ahumuda, A. J., and Watson, A. B. (1993b). An improved detection model for DCT coefficient quantization. *Proc. SPIE Human Vision, Visual Processing, and Digital Display VI*, 1453:191–201.

[Podilchuk and Zeng, 1998] Podilchuk, C. and Zeng, W. (1998). Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications*, 16:525–539.

[Portilla and Simoncelli, 2000] Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *Int'l Journal of Computer Vision*, 40:49–71.

[Ramasubramanian et al., 1999] Ramasubramanian, M., Pattanaik, S. N., and Greenberg, D. P. (1999). A perceptual based physical error metric for realistic image synthesis. *Computer Graphics (SIGGRAPH 99 Conference Proceedings)*, 33(4):73–82.

[Rane et al., 2003] Rane, S. D., Sapiro, G., and Bertalmio, M. (2003). Structure and texture filling-in of missing image blocks in wireless transmission and compression applications. *IEEE Trans. Image Processing*, 12:296–303.

[Ribas-Corbera and Lei, 1999] Ribas-Corbera, J. and Lei, S. (1999). Rate control in dct video coding for low-delay communications. *IEEE Transactions on Circuits and Systems for Video technology*, 9:172–185.

[Robson, 1966] Robson, J. (1966). Spatial and temporal contrast sensitivity functions of the visual system. *J. Opt. Soc. Am* , 56:1141–1142.

[Safranek and Johnston, 1989] Safranek, R. J. and Johnston, J. D. (1989). A perceptually tuned sub-band image coder with image dependent quantization and post-quantization data compression . *Proc. IEEE Int. Conf., Acoust., Speech, Signal Process.,* , 3:1945–1948.

[Santa-Cruz and Ebrahimi, 2000] Santa-Cruz, D. and Ebrahimi, T. (2000). An analytical study of jpeg 2000 functionalities. In *Proceeding of the International Conference on Image Processing, 2000*, volume 2, pages 49–52, Vancouver, BC, Canada.

[Sarnoff, 2003] Sarnoff (2003). Sarnoff JND vision model. *Contribution to IEEE G-2.1.6 Compression to Advanced Video Coding, Sarnoff Corp.*

[Sinha and Tewfik, 1993] Sinha, D. and Tewfik, A. (1993). Low bit rate transparent audio compression using adapted wavelets. *IEEE Transactions on Signal Processing*, 41(12):3463 – 3479.

[Skodra et al., 2000] Skodra, A., Christopoulos, C., and Ebrahimi, T. (2000). The upcoming still image compression standard. In *Proceeding of the 11th Portuguese Conference on Pattern Recognition*, pages 359–366, Porto, Portugal.

[Taubman and Marcellin, 2002] Taubman, D. and Marcellin, M. (2002). JPEG2000: Image Compression Fundamentals, Standards and Practice . *Kluwer Academic Publishers* .

[Tong and Venetsanopoulos, 1998] Tong, H. Y. and Venetsanopoulos, A. N. (1998). A perceptual model for jpeg applications based on block classification, texture masking, and luminance masking. *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 3.

[Tu et al., 2006] Tu, Y., Yang, J., and Sun, M. (2006). Efficient Rate-Distortion Estimation for H.264/AVC Coders. *IEEE Transactions on Circuits and Systems for Video technology*, 16:600–611.

[van den Branden Lambrecht and Kunt, 1998] van den Branden Lambrecht, C. J. and Kunt, M. (1998). Characterization of human visual sensitivity for video imaging applications. *Signal Process.*, 67(3):255–269.

[Vistex, 2002] Vistex (2002). Vistex texture database. Available: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

[Wang et al., 2006] Wang, J., Tong, X., Lin, S., and et al. (2006). Appearance manifolds for modeling time-variant apperance of materials. *ACM Trans. Graph.*, 25:754–761.

[Wang et al., 2002] Wang, Y., Ostermann, J., and Zhang, Y.-Q. (2002). Video processing and communications. *Prentice Hall* .

[Wang et al., 2004] Wang, Z., Lu, L., and Bovik, A. (2004). Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication, special issue on objective video quality metrics*, 19(2).

[Watson, 1993] Watson, A. (1993). DCTune: a technique for visual optimization of DCT quantization matrices for individual images. *Soc. Inf. Display Digest Tech. Papers XXIV* , pages 946–949.

[Watson et al., 1997] Watson, A., Yang, G., Solomon, J., and Villasenor, J. (1997). Visibility of wavelet quantization noise. *IEEE Transactions on Image Processing* , 6:1164–1175.

[Wei et al., 2008a] Wei, L., Han, J., Zhou, K., Guo, B., and Shum, H. (2008a). Inverse texture synthesis. In *proceedings of ACM SIGGRAPH 2008.*

[Wei and Levoy, 2000] Wei, L.-Y. and Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. In *proceedings of ACM SIGGRAPH 2000*, pages 479–488.

[Wei and Ngan, 2006] Wei, Z. and Ngan, K. N. (2006). A fast macroblock mode decision algorithm for H.264. In *Proc. of IEEE Asia-Pacific Conf. on Circuits and Syst. (APCCAS) 2006*, Singapore.

[Wei and Ngan, 2007] Wei, Z. and Ngan, K. N. (2007). A fast rate-distortion optimization algorithm for H.264/AVC. In *Proc. of IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing'07 (ICASSP'07)*, Honolulu, USA.

[Wei and Ngan, 2008a] Wei, Z. and Ngan, K. N. (2008a). The perceptually transparent coding for image. *IEEE Trans. Circuits Syst. Video Technol., Submitted for publication.*

[Wei and Ngan, 2008b] Wei, Z. and Ngan, K. N. (2008b). Spatial just noticeable distortion profile for image in dct domain. *Proc. IEEE Int'l Conf. on Multimedia and Expo (ICME).*

[Wei and Ngan, 2008c] Wei, Z. and Ngan, K. N. (2008c). Spatio-temporal just noticeable distortion profile for grey scale image/video in dct domain. *IEEE Transactions on Circuits and Systems for Video Technology, Accepted for Publication.*

[Wei and Ngan, 2008d] Wei, Z. and Ngan, K. N. (2008d). A temporal just-noticeable distortion profile for video in dct domain. In *Proc. of IEEE Int'l Conf. on Image Processing '08 (ICIP'08)*, San Diego, USA.

[Wei and Ngan, 2009] Wei, Z. and Ngan, K. N. (2009). The perceptually transparent coding for image. *IEEE Int'l Symp. on Circuits and Syst. (ISCAS'09), Submitted for publication.*

[Wei et al., 2007a] Wei, Z., Ngan, K. N., and Li, H. (2007a). An efficient intra mode delection algorithm for H.264 based on fast edge classification. In *Proc. of IEEE Int'l Symp. on Circuits and Syst. (ISCAS'07)*, New Orleans, USA.

[Wei et al., 2008b] Wei, Z., Ngan, K. N., and Li, H. (2008b). An efficient intra mode selection algorithm for h.264 based on edge classification and rate-distortion estimation. *Signal Processing: Image Communication, Accepted for Publication.*

[Wei et al., 2007b] Wei, Z., Tang, K., and King N, N. (2007b). Implementation of h.264 on mobile device. *IEEE Trans. Consumer Electronics*, 53(3):1109 – 1116.

[Wei et al., 2008c] Wei, Z., Wu, F., and King N, N. (2008c). Image compression by inverse and forward texture synthesis. *IEEE Trans. Circuits Syst. Video Technol. Submitted for publication.*

[Wei et al., 2009] Wei, Z., Wu, F., and King N, N. (2009). Image compression by inverse and forward texture synthesis.

[Wei and Zhang, 2004] Wei, Z. and Zhang, X. (2004). New full pixel and sub-pixel motion vector search algorithm for fast block-matching motion estimation in H.264. In *Proc. of Int'l Conference on Image and Graphics (ICIG) 2004*, Hong Kong.

[Weinberger et al., 1996] Weinberger, M. J., Seroussi, G., and Sapiro, G. (1996). Loco-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm. In *Data Compression Conference*, pages 140–149.

[Winkler, 2000] Winkler, S. (2000). Vision models and quality metrics for image processing applications. Lausanne, Switzerland: Ecole Polytecnique Federale De Lausanne (EPFL), Swiss Federal Inst. of Technol., thesis 2313.

[Wolfgang et al., 1999] Wolfgang, R., Podilchuk, C., and Delp, E. (1999). Perceptual watermarks for digital images and video. *Proceedings of the IEEE*, 87:1108–1126.

[x264, 2004] x264 (2004). downloaded from http://developers.videolan.org/x264.html.

[Xiong et al., 2007] Xiong, Z., Sun, X., Wu, F., and Li, S. (2007). Image coding with parameter-assistant inpainting. In *IEEE Int. image processing (ICIP '07)*.

[Yang and Po, 2004] Yang, C.-L. and Po, L.-M. (2004). a fast H.264 intra prediction algorithm using macroblock properties. In *Proc. IEEE Int. Conf. Image Processing (ICIP)*.

[Yang et al., 2005a] Yang, X., Lin, W., Lu, Z., Lin, X., Rahardja, S., Ong, E., and Yao, S. (2005a). Rate control for videophone using local perceptual cues. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(4).

[Yang et al., 2005b] Yang, X., Lin, W., Lu, Z., Ong, E., and Yao, S. (2005b). Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile. *IEEE Transactions on Circuits and Systems for Video Technology* , 15:742–752.

[Yang et al., 2005c] Yang, X. K., Lin, W., Lu, Z. K., Ong, E. P., and Yao, S. S. (2005c). Just noticeable distortion model and its applications in video coding. *Signal Process.:Image Commun.*, 20:662–680.

[Yi et al., 2005] Yi, X., Zhang, J., Ling, N., and Shang, W. (2005). Improved and simplified fast motion estimation for jm, Joint Video Team(JVT) Docs. JVT-P021.

[Young, 1991] Young, R. (1991). Oh say, can you see? The physiology of vision. In *Proc. SPIE Human Vision, Visual Processing and Digital Display*, volume 1453, pages 92 – 123, San Jose, USA.

[Yu et al., 2006] Yu, A. C., Ngan, K. N., and Martin, G. R. (2006). Efficient intra- and inter-mode selection algorithms for H.264/ AVC. *Journal of Visual Communication and Image Representation*, 17:322–344.

[Zeng, 1999] Zeng, W. (1999). Visual optimization in digital image watermarking. *Workshop on Multimedia and Security at ACM Multimedia '99*.

[Zhang and Zhang, 2004] Zhang, F. and Zhang, X. (2004). Fast macroblock mode decision in H.264. In *Proc. of IEEE International Conference on Signal Processing (ICSP)*.

[Zhang et al., 2005] Zhang, X., Lin, W. S., and Xue, P. (2005). Improved estimation for just-noticeable visual distortion. *Signal Processing*, 85:795–808.