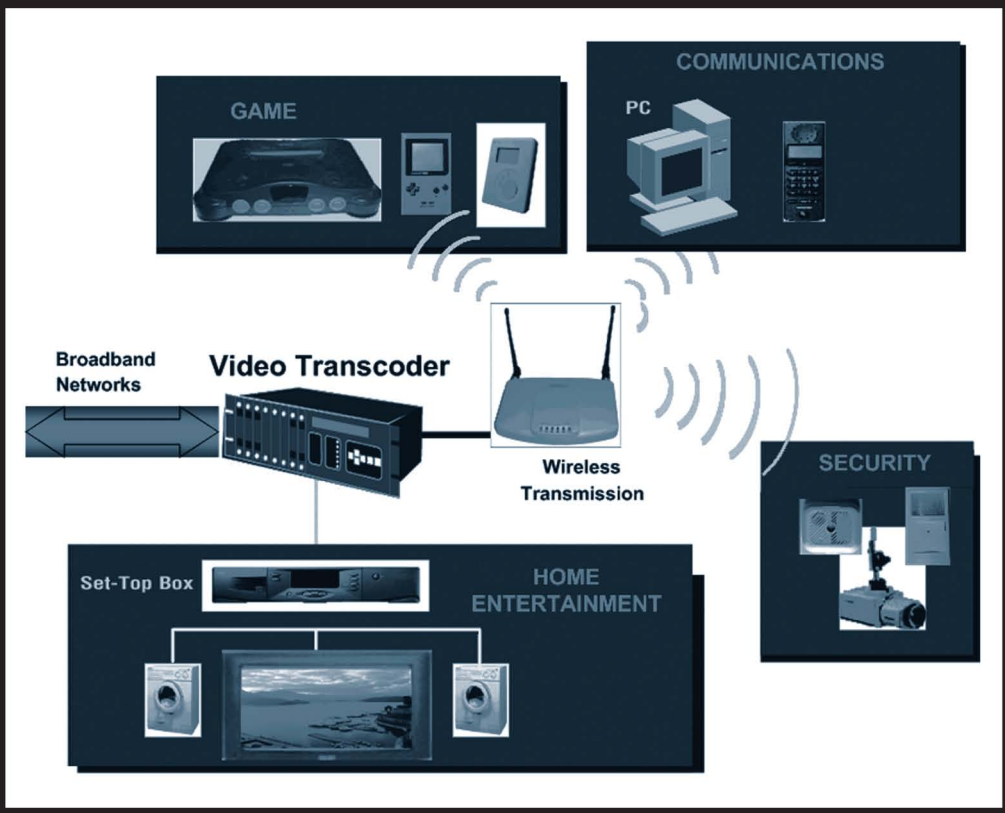
Digital Video Transcoding for Transmission and Storage



Huifang Sun • Xuemin Chen • Tihao Chiang

CRC PRESS

Digital Video Transcoding for Transmission and Storage

Digital Video Transcoding for Transmission and Storage

Huifang Sun
MERL Technology Lab, Cambridge, MA, USA

Xuemin Chen
Broadman Corporation, San Diego, CA, USA

Tihao Chiang
National Chiao-Tung University, Hsinchu, Taiwan

CRC PRESS

Boca Raton London New York Washington, D.C.

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20110713

International Standard Book Number-13: 978-1-4200-5818-5 (eBook - PDF)

Preface

During the past two decades, there has been significant progress made in digital video processing, transmission, and storage technologies. These technologies include digital video compression, digital modulation, and digital storage. Among these technologies, digital video compression is a field in which fundamental technologies are driven by practical applications. Especially, the digital video-compression standards, developed by the Moving Pictures Expert Group (MPEG) of the International Organization for Standardization (ISO) and Video Compression Expert Group (VCEG) of the International Telecommunications Union (ITU), have enabled many successful digital-video applications.

Digital video compression plays a key role in video and multimedia industries. The compression technologies bridge the gap between the huge amount of visual data required for video, multimedia transmission, and storage and limited hardware capability, despite rapid growth in the semiconductor industry. In order to effectively employ video compression technologies, many industry standards for video coding have been developed. These are MPEG-1, MPEG-2, MPEG-4, H.261, H.263, and H.264/MPEG-4 Advanced Video Coding (AVC). Several excellent books have been published on the subject of video compression and standards. These books focus on basic video coding theory and/or the video compression standards. However, there is a need for a book that provides the explanations on the standards and at the same time explains how to convert the compressed information between standards. This is the hot topic investigated by many researchers from academia and industry; our book will meet this need and will provide some of the theories and principles of video compression and transcoding technologies.

The concepts of digital video coding are in many cases sufficiently straightforward to avoid highly theoretical mathematics. The reader will find that the primary emphasis of the book is on digital video transcoding techniques, not on the basic theories and principles of digital video coding and its standards. In fact, much of the material covered is summarized via examples of practical methods for transcoder implementation, and almost all of the video transcoding technologies introduced are related to practical applications.

This book has arisen from the authors' research results from many years in both industrial and academic societies. We would like to share our experiences and research results with colleagues in the field of video transcoding. The book takes a structured approach to transcoding technologies, starting with the basic video transcoding concepts and working gradually toward the most sophisticated transcoding systems. The practical applications of transcoders are described throughout the book. The materials are summarized from many research papers, lectures, and presentations on video compression and transcoding technologies. The text is intended to be a senior/graduate level textbook and reference book for researchers and engineers in this field. We hope that college students and engineers in the video

and multimedia industry can benefit from the information in this, for the most part, self-contained text on video transcoding technologies.

The book is organized as follows. It consists of 11 chapters grouped into four parts. The first part includes two chapters, which provide the background of video coding theory, principles of video transmission, and an overview of video coding standards. The second part includes three chapters that provide the theory of video transcoding and practical problems. The third part includes three chapters and presents buffer management, packet scheduling, and encryption in the transcoding. The final part contains three chapters and describes the application of transcoding, universal multimedia access with emerging standard MPEG-21, and the end-to-end test bed. These topics occur in very recent results of research and should be welcome by the readers.

Every course has as its first lecture a sneak preview and overview of the fundamental techniques and tools useful for the following chapters. Chapter 1 provides such information. It provides fundamental concepts and theoretical basis for digital video compression techniques including fundamentals of information theory and various source coding techniques. It also covers some fundamentals for video coding such as redundancy removal using spatial, temporal, and frequency domain techniques.

In Chapter 2, the topics of digital video coding standards are discussed. First, the general principles and structures of digital video coding are presented. Then the basic tools for video coding standards are introduced. In order to improve coding efficiency and increase functionality, several enhancement tools for digital video coding standards are introduced and the basic ideas behind these tools are described. Finally, a summary of different digital video coding standards and several encoding issues are presented.

Chapter 3 provides the theoretical basis and fundamentals for the video transcoding. The general concept and possible applications for video transcoding are first introduced. Then, various transcoding architectures for specific purposes are then presented and compared. Various transcoding algorithms are developed and analyzed. Finally, an alternative to achieve transcoding, namely Fine Granularity Scalabiliy, is presented.

Chapter 4 focuses on transcoding performance optimization for various functionality. The first topic is the performance improvement for reduced spatial resolution transcoding. The second topic focuses on the temporal resolution adaptation. The third topic concentrates on the syntactical adaptation for different formats of bit streams such as MPEG-1/2/4 and H.264. Lastly, several topics such as error resilient transcoding, logo insertion, watermarking, switched pictures and picture-in-picture transcoding are covered. The complexity analysis for various transcoding architectures is also provided.

In Chapter 5, transport-level transcoding techniques are introduced. To explain transport-level transcoding, we first present the basic concept of MPEG-2 system. MPEG-2 systems have two specified stream formats: transport and program, which are specified for a set of different applications. We present the transcoding techniques, which are used to perform the conversions between the transport stream and the program stream.

In Chapter 6, we present video synchronization techniques, such as system clock recovery and time stamping for decoding and presentation. As an example, we use

MPEG-2 transport systems to illustrate the key function blocks of this video synchronization technique. In MPEG-2 systems, clock recovery is possible by transmitting time stamps called *program clock references* (PCRs) in the bit stream. The PCRs are generated at the encoder by sampling the system time clock (STC). Since the decoder's free-running system clock frequency does not exactly match the encoder's STC, the reference time is reconstructed by means of a phase-locked loop (PLL) and the received PCRs. The encoder's STC of a video program is also used to create time stamps that indicate the presentation and decoding timing of video. Methods for generating the decoding and presentation time stamps in the video encoder are discussed in this chapter. In particular, the time-stamping methods for MPEG-2 video are introduced as examples. These methods can be directly used in other compressed video.

In Chapter 7, constraints on video compression/decompression buffers and the bit rate of a compressed video bit stream have been discussed. The transmission channels impose these constraints on video encoder/transcoder buffers. First, we introduce concepts of compressed video buffers. Then conditions that prevent the video encoder/transcoder and decoder buffer overflow or underflow are derived for the channel that can transmit a variable bit rate video. Next, the discussion focuses on analyzing buffer, timing recovery, and synchronization for video transcoder. The buffering implications of the video transcoder within the transmission path have also been analyzed.

In the applications of transcoding, digital right management and security issues become more and more important. In Chapter 8, we study the basic security concepts, algorithms, and models for transporting of compressed digital video. We describe a number of important cryptographic algorithms that are used to protect high-value video content and review several configurations of conditional access systems for digital video transport systems. In particular, we outline the DES algorithm in details. We introduce four modes of operation and their applications in both block cipher and stream cipher. We also look at the area of public-key cryptosystems and examine two algorithms, the Rivest, Shamir, and Adleman encryption algorithm, based on the product of two large prime numbers, and the Diffie-Hellman key exchange algorithm, based on the discrete logarithm problem for finite field. We also review three configurations of conditional access schemes. Finally, we have introduced some basic ideas of the multi-hop encryption systems. We believe this chapter will be very useful for many researchers and engineers for developing the transcoding techniques in many applications.

Chapter 9 is devoted to the application and implementation of video transcoding. The first application is the transcoder of MPEG-2 to MPEG-4 bit streams. The MPEG-4 video coding standard has been used for video streaming and mobile terminals, but much content in video servers is encoded with MPEG-2; therefore, we need convert the MPEG-2 to MPEG-4 for users with MPEG-4 available decoder to receive the MPEG-2 encoded contents. Second, the techniques of error resilience transcoding, which are related to video transcoder for IP network, are discussed. In the applications of IP networks and wireless networks, the function of error resilience is important. How to increase the error robustness for the transcoded bit stream is introduced. Finally, the object-based transcoding technique for MPEG-4 will be presented.

In Chapter 10, we address several transcoding aspects related to the distribution of digital content. The first objective is to provide an overview of the universal multimedia access (UMA) concept. The primary function of UMA services is to provide the best quality of service or user experience by either selecting/adapting the content format to meet the playback environment, or adapting the content playback environment to accommodate the content. The second objective is to describe how the concept of UMA relates to the emerging MPEG standard, Digital Item Adaptation (DIA), which is the Part 7 of the MPEG-21 standard. The update on the standards activity in this area is presented. Finally, we address the impact that DIA will have on transcoding strategies, and we analyze some areas of future research.

In Chapter 11, we introduce the concept of real-time transport protocol and carriage of multimedia content over IP networks. The first section covers the various elements of the video streaming and transcoding system. The second section describes how content can be carried over IP networks. Details of the real-time transport protocol will be described. An implementation example of such a system is described. The last section describes some simulation results and conclusions.

Acknowledgments

We wish to acknowledge everyone who helped in the preparation of this book—in particular, the reviewers, who have made detailed comments to guide us in our final choice of content. We also wish to acknowledge Nora Konopka, the Engineering and Environmental Sciences editor at CRC Press, who helped in many aspects and made very efficient arrangements for publishing this book.

The first author would like to express his deep appreciation to his colleague, Dr. Anthony Vetro, for fruitful technical discussions related to some contents of this book and many joint publications and patents on this topic, which are reflected in this book. He would like to thank Drs. Ajay Divakaran, Fatih Porikli, Hao-Song Kong, Zafer Sahinoglu, and Jun Xin for their help in many aspects on this book. He also would like to acknowledge Drs. Richard Waters, Takashi Kan, Kent Wittenburg, and Joe Marks for their continuing support and encouragement. His acknowledgement also goes to Drs. Tokumichi Murakami and Kohtaro Asai for their friendly support and encouragement. He also would to thank Professor Wen Gao and his students Yan Lu and Lujun Yuan for their help.

Firstly, the second author wishes to thank his thesis advisor, Professor Dimitris Anastassiou, who introduced him into the field of digital video compression while it was still an emerging area. His guidance opens the door and the second author could not be here without his help. He also wishes to thank his parents, Lian-Yuan Chiang and Shing Yeh, who have been very supportive during this project. He also wishes to thank Dr. Ya-Qin Zhang for his leadership and friendship in completing this book. The second author wishes to extend special thanks to Professors Che-Ho Wei, Suh-Yin Lee, and David Lin for their help. The second author wishes to thank Professors Hsueh-Ming Hang and Chun-Jen Tsai for their support and help in this project. Professor Hang has been extremely supportive since the second author joined National Chiao Tung University in 1999. This project would not be possible without his help. The second author wishes to thank Professor Sheng-Jyh Wang for his help in this project and his continuing friendship. He would like to thank Dr. Chung-Neng Wang for his friendship and assistance in the writing of this book. He wishes to thank the group members, including Peter Chuang, Yao-Chung Lin, Chih-Hung Lee, and Hsiang-Chun Huang at the Multimedia Communications Group of the Communications and Signal Processing Laboratory at National Chiao Tung University in Taiwan. The second author wishes to thank Dr. Jun Xin, Prof. Chia-Wen Lin and Prof. Ming-Ting Sun for providing an early draft of their tutorial on transcoding. He wishes to thank Dr. Anthony Vetro for his helpful discussions on the various transcoding issues in one of the joint works. He wishes to thank Mr. Shinga Chen for his help.

The third author would like to thank Dr. Wade Wan for his review of parts of the manuscript and for his thoughtful comments. He also gratefully acknowledges Robert Eifrig, Dr. Ajay Luthra, Dr. Fan Ling, Dr. Vincent Liu, Dr. Sam Narasimhan, Dr. Krit Panusopone, Dr. Ganesh Rajan, and Dr. Limin Wang for their contributions

in many joint patents, papers, and reports that are reflected in this book. He would also like to thank Professors Irving S. Reed, Tor Helleseth, Weiping Li, Ronald Crochiere, Homer H. Chen, Ya-Qin Zhang, Guirong Guo, T. K. Truong, T. C. Cheng, and Dr. Anthony Vetro for their continuing support and encouragement.

Support for the completion of the manuscript has been provided by the National Chiao-Tung University (Taiwan) and to all we are truly grateful. In particular, we truly appreciate the attentiveness that Peter Chuang has given to the preparation of the manuscript. We also wish to thank Chih-Hung Lee for his kind and timely assistance in the production phase of the project. Authors also would like to express their appreciation to many friends and colleagues of the MPEGers who provided wonderful MPEG documents and tutorial materials that are cited in many chapters of this book. It was important to be able to use many published results in the text. We would like to thank the people who made possible these important contributions.

Finally, we would like to show our great appreciation to our families for their constant help, support, and encouragement.

About the Authors

Huifang Sun is the vice president of Mitsubishi Electric Research Laboratories (MERL), and the deputy director of technology laboratory of MERL. He is a MERL Fellow and an IEEE Fellow. He graduated from Harbin Military Engineering Institute, Harbin, China, and received the Ph.D. from University of Ottawa, Canada. He was an Associate Professor at Engineering Department of Fairleigh Dickinson University before moving to Sarnoff Corporation in 1990 as a member of technical staff, and was promoted to a technology leader of Digital Video Communication later. In 1995, he joined MERL. His research interests include digital video/image compression and digital communication. He has published a textbook and more than 120 journal and conference papers. He holds 30 U.S. patents and has more pending in the area of digital video compression, processing, and communications. He received AD-HDTV Team Award in 1992 and Technical Achievement Award for optimization and specification of the Grand Alliance HDTV video compression algorithm in 1994 at Sarnoff Lab. He received the best paper award of 1992 for *IEEE Transactions on Consumer Electronics*, the best paper award of 1996 ICCE, and the best transactions paper award of 2003 *IEEE Transactions on Circuits and Systems for Video Technology*. He is now an Associate Editor for *IEEE Transactions on Circuits and Systems for Video Technology* and the Chair of Visual Processing Technical Committee of the IEEE Circuits and System Society.

Xuemin Chen is a technical director of Broadband Communications Business Group of Broadcom Corporation and an IEEE fellow. He has a Ph. D. degree in electrical engineering from University of Southern California. He had held various engineering positions such as research scientist, senior staff/manager, and senior principal scientist in American Online (Johnson & Grace), General Instrument Corporation (currently the Motorola Broadband Communication Sector), and Broadcom, respectively. He has published two graduate-level textbooks on digital communication, entitled *Error-Control Coding for Data Network* and *Transporting Compressed Digital Video*. He is an inventor of more than 70 granted or published patents worldwide in digital image/video processing and communication. He has also published over 60 research articles and contributed many book chapters in data compression and channel coding. He has served technical committees of various conferences on signal processing. He has also served as an associate editor of *IEEE Transactions on Circuit and Systems for Video Technology*, from 2000 to 2004. He actively involved in developing ISO MPEG-2 and -4 standards.His research interests include information theory, digital video compression and communication, computer architecture, VLSI system-on-a-chip, error-control coding, and data networks. His applied works concentrate on design and implementation of broadband communication system architectures, digital television systems, video transmission over cable and satellite and IP networks, and media-processor/DSP/ASIC architectures.

Tihao Chiang was born in Cha-Yi, Taiwan, Republic of China, in 1965. He received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1987, and the M.S. degree in electrical engineering from Columbia University in 1991. He received his Ph.D. degree in electrical engineering from Columbia University in 1995. In 1995, he joined David Sarnoff Research Center as a member of technical staff. Later, he was promoted as a technology leader and a program manager at Sarnoff. While at Sarnoff, he led a team of researchers and developed an optimized MPEG-2 software encoder. For his work in the encoder and MPEG-4 areas, he received two Sarnoff achievement awards and three Sarnoff team awards. Since 1992 he has actively participated in ISO's Moving Picture Experts Group (MPEG) digital video coding standardization process with particular focus on the scalability/compatibility issue. He is currently the co-editor of the part 7 on the MPEG-4 committee. He has made more than 90 contributions to the MPEG committee over the past 10 years. His main research interests are compatible/scalable video compression, stereoscopic video coding, and motion estimation. In September 1999, he joined the faculty at National Chiao-Tung University in Taiwan, R.O.C. Dr. Chiang is currently a senior member of IEEE and holder of 13 U.S. patents and 30 European and worldwide patents. He was a co-recipient of the 2001 best paper award from the IEEE *Transactions on Circuits and Systems for Video Technology.* He has published over 50 technical journal and conference papers in the field of video and signal processing. He was a guest editor for *IEEE Transactions on Circuits and Systems for Video Technology* and is the Chair of Visual Processing Technical Committee of IEEE Circuits and System Society.

Contents

1 Fundamentals of Digital Video Compression

In this chapter, we introduce the fundamentals of digital video compression. The content will include the following topics. We will discuss the concept of entropy, which pertains to the minimum number of bits necessary to carry information without any loss.

Rate distortion theory describes the fundamental bounds for compression but does not offer any specific methods for the implementation of such bounds. In practice, several lossless encoding techniques are used for compression standards. In particular, variable length codes are used to achieve compression when the more probable event is represented with shorter code and vice versa. There are several types of variable length code, but we will limit our discussion to Huffman code, Golomb code, and arithmetic code, which are commonly used in the international standards, such as MPEG, JPEG, and ITU standards.

To represent the picture in full color, the scene is typically captured and digitized as an RGB color space representation. We will review commonly used color space representations; each approach has its advantages and disadvantages for different purposes. However, such a color representation possesses significant redundancy among components, which makes it unsuitable for compression purposes. Therefore, RGB is rarely used for video compression standards, such as MPEG and ITU. For compression purposes, the removal of the spectral redundancy associated with each color component is the first step achieved with the YUV color representation. In addition to spectral redundancy, there are two other inherent redundancies, in the spatial and temporal domains. The spatial redundancy comes from the correlations between neighboring pixels within a frame. The temporal redundancy comes from the correlations between consecutive frames in a video. We will describe various techniques how such redundancies can be exploited.

To fully exploit spatial and temporal redundancy, there are practical approaches to achieve a feasible implementation. In particular, spatial redundancy is removed with spatial domain transformation into the frequency representation, such as with the discrete cosine transform (DCT), discrete wavelet transform (DWT), and subband decomposition. We will review several fast algorithms for such transformations. As for temporal redundancy removal, the typical approach used in video is block-based motion estimation and compensation. Block-based motion estimation can be implemented efficiently with existing VLSI technology. The significant advances of VLSI technology enable us to consider more complicated algorithms to achieve further compression. Thus, variable block size motion estimation and compensation is now accepted and used in state-of-the-art video standards, such as H.264/AVC.

1.1 FUNDAMENTALS OF INFORMATION THEORY

The origin of information theory can be traced back to the landmark paper "A Mathematical Theory of Communication," by Claude E. Shannon in 1948 [1-22]. The theory is now referred to as "The Mathematical Theory of Communication."

Let's consider a simple type of information — binary information source — since the binary representation of information is commonly used in data storage and computers. For example, the decimal digit is typically recorded with natural binary representation as shown in Table 1.1. We see three ways to encode the four symbols s_i in the table. The first approach is to use natural binary code to represent the information source ***S***. The advantage is its ease of interpretation and implementation of arithmetic operations, such addition and subtraction. However, natural binary representation is not the only way to store these symbols. For example, Gray code presents another method of storing the information source. The Gray code possesses only single-bit change between consecutive symbols, which is useful for achieving error protection and correction. However, both binary and Gray codes denote the symbols with an equal number of bits per symbol. Both codes are referred to as *fixed length code*.

The third approach uses code words with different lengths to represent the information source ***S.*** As shown in Table 1.1, the probability for each symbol is different, so it is obvious that a more frequent symbol should be denoted with a shorter code word such that the average length is smaller. With this new code, the average number of bits used to represent each symbol is reduced from two bits to

$$\left(\frac{1}{2}\times 1+\frac{1}{4}\times 2+\frac{1}{8}\times 3+\frac{1}{8}\times 3\right)=1.75$$

The fundamental question is to determine the minimal number of bits to represent an information source with known statistics. The pursuit of the answer to such a question leads to the concept of entropy.

1.1.1 Entropy

The study of entropy starts with the understanding of the measure of information. Intuitively, the identification of a less probable symbol carries more information if such a symbol occurs. That is, an unusual event should carry more information. For example, the information on the current weather in Los Angeles has a higher probability of sunshine

TABLE 1.1
Fixed Length Code and Variable Length Code

Symbol	Probability	Natural Binary Code	Gray Code	Variable Length Code
s_1	0.5	00	00	1
s_2	0.25	01	01	01
s_3	0.125	10	11	000
s_4	0.125	11	10	001

than snow. Therefore, a snow day event will carry more information because it is less likely. The event that the sun rises in the east carries no information since it will surely occur. To implement such a concept, we define the "measure of information" as follows.

Definition 1.1 *Let the symbol s_i occur with probability p_i. The measure of information is defined by $I(s_i) = -\log p_i$ units of information. If the base of the logarithm is 2, the resulting unit of information is called a* binary unit*, or* bit.

The measure of information $I(s_i)$ is defined as $-\log p_i = \log(1/p_i)$. As p_i becomes smaller, the information it carries becomes larger. Based on the measure of information, the average of bits necessary to represent can be computed as an expected or average value of the measure of information defined as follows.

Definition 1.2 *Let the symbol s_i occur with probability p_i. The average amount of information source of symbol s_i can be computed as the entropy $H(S)$ of the source as defined by*

$$H(S) \equiv -\sum_{i=1}^{N} p_i \log p_i.$$

In his paper, Shannon proved that the defined entropy is indeed the minimal number of bits necessary to encode the information source.

Theorem 1.1 *[Source Coding Theorem] A source with entropy H can be encoded with arbitrarily small error probability at any rate as long as $R > H$.*

However, there is no unique way to achieve such a bound. Thus, it is natural to investigate how such bounds can be achieved. Before we answer that question, we want to ask another: what is the effect of the probability distribution on the entropy? We know that $\sum_{i=1}^{N} p_i = 1$ but there are infinite combinations p_i and we want to determine the maximal entropy in representing this source and under what conditions. The following theorem answers such a question without proof.

Theorem 1.2 *For a memoryless information source with N symbols source alphabet $\{s_i\}$, the maximum of the entropy is exactly* $\log(N)$*. Such maximum is achieved if and only if all the symbols are equiprobable.*

Theorem 1.1 illustrates that the entropy is maximized when the probabilities p_i satisfy the following condition: $p_1 = p_2 = \cdots = p_N = 1/N$. Obviously, the entropy is computed as an average of information

$$\begin{aligned} H(S) &\equiv -\sum_{i=1}^{N} p_i \log p_i \\ &= -\underbrace{\left[\frac{1}{N}\log\left(\frac{1}{N}\right) + \cdots + \frac{1}{N}\log\left(\frac{1}{N}\right)\right]}_{N} \\ &= \log N \end{aligned}$$

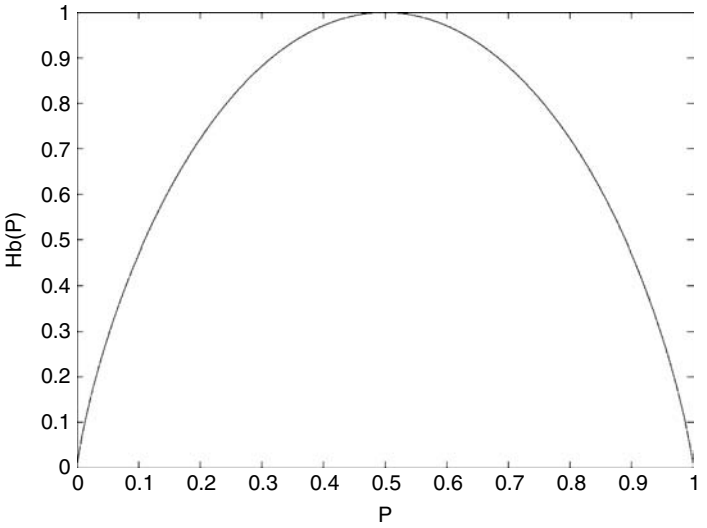


FIGURE 1.1 Entropy for the probability [1-22].

Example 1.1 *For a binary memoryless information source with probabilities* ***p*** *and* $1 - \boldsymbol{p}$, *we have the entropy* $H_b(X) = -p \log p - (1 - p)\log(1 - p)$. *This function is known as binary entropy function as shown in Figure 1.1. There are several interesting properties that can be observed in this example.*

The maximum of entropy occurs when $p = 0.5$. This means that the entropy is maximal when both symbols are equiprobable and thus the uncertainty is maximal. The minimum of entropy occurs when $p = 0$ or 1, which means that no information of the one event is certain and the other event is impossible. Such a scenario requires no information.

1.1.2 Properties of Block Codes

In order to implement a practical system, specific code words need to be assigned for each symbol. We now define the block code since both Huffman and Golomb codes belong to this category.

Definition 1.3 *A block code is a code that maps each symbol* s_i *of the information source* ***S*** *into a fixed sequence of the code alphabet.*

The first two codes in Table 1.2 are both block codes because each symbol s_i of the information source ***S*** is mapped to a fixed sequence of code alphabet. However, each sequence need not be distinct. It is obvious that indistinct code words result in confusion at the receiver side. Thus, we need to add another property for the block code, called *nonsingularity*, to remove such confusion. A nonsingular code is defined as follows.

TABLE 1.2
Nonsingular Block Codes

Symbols	Code A	Code B
s_1	0	1
s_2	10	10
s_3	110	11
s_4	1110	111

Definition 1.4 *A code is* ***nonsingular*** *if every element of the range of the source symbols maps into different code words.*

For nonsingular code, the decoder can decode each symbol if the symbols are properly separated. However, this may not be true when symbols are concatenated. For example, the sequence "111" may come from a sequence of symbols including "$s_1s_1s_1$," "s_3s_1," "s_1s_3" or "s_4." Thus, nonsingular block code does not guarantee that the symbols are decodable when multiple symbols are strung together. In information theory, such an action of concatenation is referred to as *extension*, defined as follows.

Definition 1.5 *An* $\boldsymbol{n}^{\text{th}}$ *extension of a code is a mapping from a finite length of* $\boldsymbol{n}$ *symbols to a finite length of source symbols* s_i*. A code is uniquely decodable if its* $\boldsymbol{n}^{\text{th}}$ *extension is nonsingular for all finite* $\boldsymbol{n}$**.**

As shown in Table 1.3, the second extension of code B has mapped both "s_3s_1" and "s_1s_3" to the same sequence, "111." The decoder cannot distinguish between these two cases, so Code ***B*** is not uniquely decodable.

In addition to unique decodability, it is desirable to have the ability to decode as soon as the full code word is received, which is referred to as *instantaneous decoding*. As shown in Table 1.4, code ***D*** is not instantaneous because the decoding of "100" from the sequence "1001" requires the fourth bit, "1," to make a decision that the first three bits represent the symbol s_3. To achieve instantaneous decoding, the following theorem states the condition as the prefix code requirements.

TABLE 1.3
Second Extension of Code B

Symbols	Code C	Symbols	Code C
s_1s_1	11	s_2s_1	101
s_1s_2	110	s_2s_2	1010
s_1s_3	111	s_2s_3	1011
s_1s_4	1111	s_2s_4	10111
s_3s_1	111	s_4s_1	1111
s_3s_2	1110	s_4s_2	11110
s_3s_3	1111	s_4s_3	11111
s_3s_4	11111	s_4s_4	111111

TABLE 1.4
Noninstantaneous Code

Symbols	Code D
s_1	1
s_2	10
s_3	100
s_4	1000

Theorem 1.3 *A code is a prefix code or instantaneous code if and only if no codeword is a prefix of any other codeword.*

1.2 VARIABLE LENGTH CODE

There are several types of variable length code. Due to its simplicity, the most commonly used code is Huffman code, which has been adopted in JPEG, MPEG, and several other standards. With given statistics, one can show that Huffman code is indeed optimal. However, the Huffman code requires a predetermined probability distribution so that it can provide an optimal code to match the statistics. If this is not the case, the code table is no longer optimal. The disadvantage can be solved using several Huffman codes for specific probability distributions or statistics.

However, there is another limitation with Huffman code, which is the need for constructing a table at both the encoder and decoder. It becomes a problem when the number of symbols is significant. Golomb code can be used to address such a problem assuming a geometrically distributed signal. The main advantage of Golomb code is its simplicity. The Golomb code does not require any code table and uses a very simple decoding method ideal for hardware implementation. It also allows easy adaptation to the statistics by selecting appropriate Golomb code. Furthermore, the Golomb code can be extended for an infinite number of source symbols. Golomb code has been used for the JPEG-LS and H.264/AVC standards.

Both Huffman and Golomb codes are block codes. Another important type of source coding technique involves non-block code. This non-block code is known as the *arithmetic* code, and has been used for several standards, such as JPEG and MPEG-4. One distinctive feature of arithmetic code is that it does not map a source symbol to a specific code word. The advantage of arithmetic code is that it does not assume any statistics for the source, as do the Huffman and Golomb codes. Instead, it estimates the statistics on the fly and uses the newly estimated statistics to improve the coding efficiency. Such a property makes it suitable to adapt to statistically nonstationary source signals, such as video and image. Another feature is the possibility of using context to classify the source signals. This allows similar improvement attained with multiple Huffman code tables without the complexity. Despite its multiple advantages, the arithmetic code suffers vulnerability to transmission noise. Error resilience is difficult to achieve for arithmetic code.

TABLE 1.5
Noninstantaneous Code

Symbols	Probability	Code word
s_1	1/2	0
s_2	3/16	11
s_3	3/16	100
s_4	2/16	101

In the following three sections, we will describe the three variable length codes typically used in video coding standards.

1.2.1 Huffman Coding

Huffman code is the optimal prefix code in terms of shortest expected length for a given distribution.

1. Sort the source symbols according to the probability.
2. Merge the two least probable symbols.
   a. Assign the top branch "0".
   b. Assign the bottom branch "1".
3. Check whether the number of symbols left is 2. If this is true, complete the code assignment. Otherwise, go to Step 1.

We will illustrate this algorithm with an example.

Example

Let us assume the information source has the symbols $\{s_1, s_2, s_3, s_4\}$ with the probabilities as shown in Table 1.5. As shown in Figure 1.2, the symbols s_3 and s_4 are combined to make a new symbol s_2'. Please note the tranposition of order from s_2 to s_3'. This is done to sort the symbols according to the magnitude of probabilities.

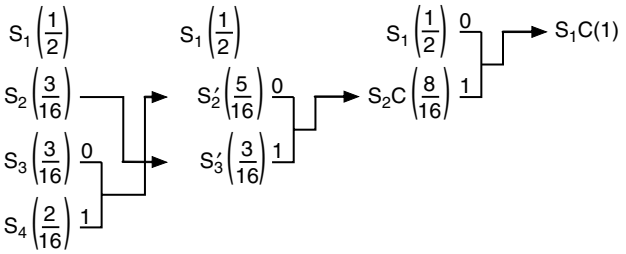


FIGURE 1.2 Huffman code construction.

TABLE 1.6
Unary Code

Symbols	Value	Probability	Code word
s_1	0	1/2	0
s_2	1	1/4	10
s_3	2	1/8	110
s_4	3	1/16	1110
⋮	⋮	⋮	111⋯ 0

1.2.2 Golomb Code

The Golomb code is a family of code that is designed with an assumed distribution that the probability will decrease as the value is increased [1-21]. An example of such a code is the unary code, illustrated in Table 1.6. Sometimes, we refer to the unary code as *comma code* because the termination symbol "0" acts likes a comma for the signal. A generalization of such code is to separate the code into two parts, where the first part is a unary code and the second part is a different code. The Golomb code was originally designed to code the run length of an event. In particular, it was illustrated as a run of success in a roulette game.

The main advantage of Golomb code is its ease of implementation, where no table is necessary for both the encoder and decoder. It also allows simple adaptation to the local statistics should the signals exhibit nonstationary behavior.

The Golomb code is parameterized with a nonzero positive integer ***m.*** To represent an integer n using Golomb code with parameter m, we need to compute two numbers, namely, q and r, where q is

$$q = floor\left(\frac{n}{m}\right) \quad \text{and} \quad r = n - qm$$

Thus, we have

$$n = qm + r$$

The quotient is coded with the unary code as described in Table 1.6. The remainder r satisfies the relationship $0 \le r < m$. Thus, it takes at most $\log_2 m$ bits for representation.

It can be shown that Golomb code with parameter ***m*** is optimal for the geometrical distribution

$$P(n) = (1-p)p^{n-1} \quad \text{and} \quad m = Q\left(-\frac{1}{\log_2 p}\right)$$

The function $Q(x)$ is the smallest integer greater than or equal to x.

Example

Let us design a Golomb code for $m = 4$. Since

$$q = floor\,(\log_2(4)) = 2$$

The Golomb code for $m = 4$ can be expressed as in Table 1.7.

TABLE 1.7
Golomb Code with Parameter 4

Value	q	Code	r	Code	Code word
0	0	0	0	00	000
1	0	0	1	01	001
2	0	0	2	10	010
3	0	0	3	11	011
4	1	10	0	00	1000
5	1	10	1	01	1001
6	1	10	2	10	1010
7	1	10	3	11	1011

1.2.3 Arithmetic Code

The Huffman code approaches the entropy *H(S)* for a given information source *S* with a set of probability distributions $\{p_i\}$. This is practically achieved with an extended alphabet, in which alphabets are concatenated to form a larger alphabet set. However, the approach becomes impractical when the $\{p_i\}$ is highly skewed and the number of alphabets is very small. In this case, the number of concatenations needed to achieve entropy may reach tens of thousand of symbols. This will make the implementation of the decoder expensive and time consuming. Furthermore, the code becomes inefficient when the probability distribution does not match the actual statistics of the signal.

The idea of arithmetic code can be considered as a mapping of a binary sequence to a number in the unit interval [0, 1). Let us consider a sequence of symbols, s_1, $s_2,s_3,s_4,\ldots,s_n$ and place a zero and decimal point before it so that we have $0.s_1,s_2$, $s_3,s_4,\ldots,s_n$." Thus, we can consider a binary sequence a real number between 0 and 1, excluding 1. A function that maps a sequence of random variables into a unit interval can be a cumulative distribution function. The arithmetic code is best illustrated with an example.

Example

Consider a binary source with an alphabet of size two {0, 1}. The set of probability is $\{\frac{3}{4},\frac{1}{4}\}$. We would like to encode the sequence {0, 0, 1, 0}

There are a few interesting observations.

- The encoder does not generate any bit for encoding the first three symbols. This is a clear distinction between arithmetic code and the Huffman and Golomb codes, where the latter use at least one bit to represent a symbol.
- Except in Huffman code, the encoder does not require any code table to generate the bit stream.
- The less probable symbol causes the encoder to generate more bits. This matches the intuition embodied in both the Huffman and Golomb codes.

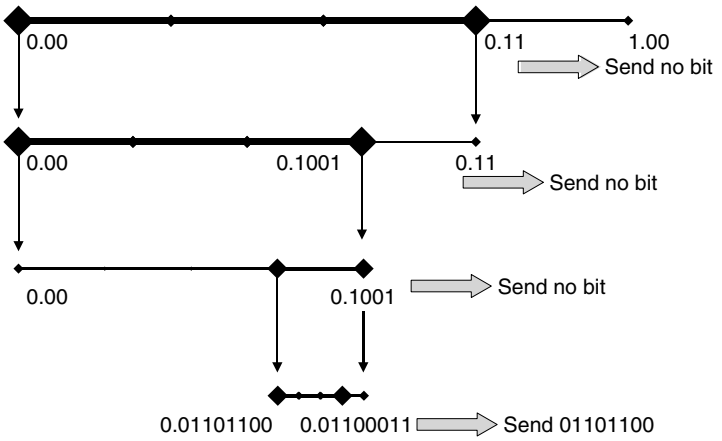


FIGURE 1.3 Arithmetic encoding of a short sequence.

- The implementation of the encoder is extremely simple. It only requires two variables storing the left and right bounds of the interval. An alternative implementation is store the left bound and the length of the interval.
- When the source is more skewed, the performance of the encoder is better; the performance gap between Huffman code and arithmetic code also widens.
- The probability model does not have to be identical for each subdivision of the interval. Thus, it is possible to update the probability model as shown in Figure 1.4. However, the encoder and decoder have to switch to the same probability model simultaneously to ensure correct decoding.

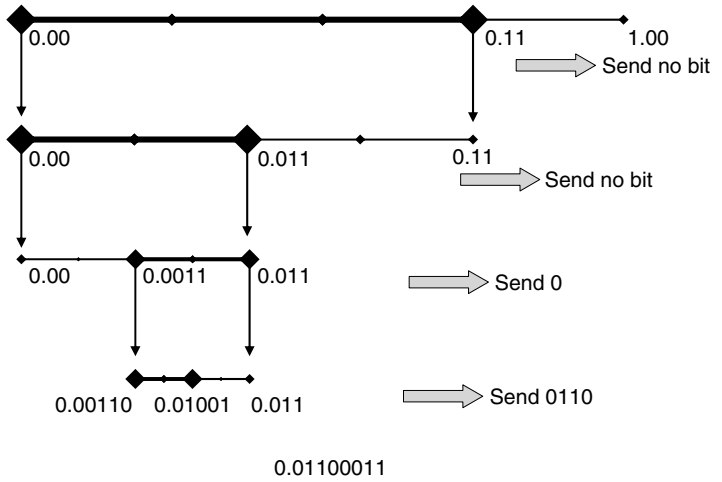


FIGURE 1.4 Arithmetic encoding of a short sequence.

1.3 Fundamentals of the Human Visual System

Color as perceived by humans can be conceptualized as a combination of the tristimuli red, green, and blue, which are called three *primary* colors. With these three primary colors, we can derive several color representations, or color spaces, using either linear or nonlinear transformations.

1.3.1 Color Space Conversion and Spectral Redundancy

The linear transformations include *YIQ*, *YUV*, and $l_1l_2l_3$ color spaces, and the nonlinear transformations include normalized RGB (Nrgb), his, and CIE spaces. The red, green, and blue components can be represented by the brightness of the object through three separate filters for each color based on the following equations.

$$R = \int_{\lambda} E(\lambda) S_R(\lambda) d\lambda$$

$$G = \int_{\lambda} E(\lambda) S_G(\lambda) d\lambda$$

$$B = \int_{\lambda} E(\lambda) S_B(\lambda) d\lambda$$

where S_R, S_G, and S_B are the color filters for the radiance $E(\lambda)$ and λ is the wavelength. The RGB color space is the most commonly used model for the television and video for consumer electronics. However, it is not frequently used for video compression because of the high correlations among each component.

Weber's law states that manipulation of colors is a linear operation, which includes either scaling or addition. Any colors can be created by the combination of the three colors and such a combination is unique. When two colors are mixed, the resultant color is the sum of the values of each color. Such a linear property enables us to perform linear transformation of color components, which can be inversed back or translated to different spaces.

The *YIQ* standard is used in American television. The transformation is defined as

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.253 & -0.312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

where $0 \le R, G, B \le 1$. The YUV standard is used in European television and international standards for digital video. The transformation is defined as

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

where $0 \le R, G, B \le 1$.

The normalized RGB is used to make the color representations independent of lighting changes. However, the dimensionality is reduced because when two components are determined, the third component can be determined.

$$r = \frac{R}{R+G+B}$$

$$g = \frac{G}{R+G+B}$$

$$b = \frac{B}{R+G+B}$$

It is obvious that $r + g + b = 1$. Another variation is defined as

$$Y = c_1 R + c_2 G + c_3 B$$

$$T_1 = \frac{R}{R+G+B}$$

$$T_2 = \frac{G}{R+G+B}$$

$$c_1 + c_2 + c_3 = 1.$$

The CIE color space has three primaries denoted as *X*, *Y*, and *Z*, which can be computed by a linear transformation from the RGB color space. The transformation matrix is defined as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0 & 0.066 & 1.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

1.4 VIDEO CODING FUNDAMENTALS

The video source consists of a sequence of frames that are temporally sampled from a three-dimensional scene at a predetermined period depending on the standards. For example, the North American standard adopts a temporal sampling frequency of 30 or 29.97 frames per second. Each frame is horizontally and vertically sampled at a progressive or interlaced sampling grid. The interlaced format is a legacy from analogue television that allows an effective temporal resolution of 60 Hz and an effective vertical resolution of 525 lines whereas the actual sampling rate is half, as shown in Figure 1.5.

The interlaced format can be considered as compression of a factor of two that fits the vast video information into a 6 MHz channel. When the analogue video information is digitized, the objective of a video coder is to compress the source as much as possible. There are three sources of redundancies including spatial, temporal, and statistical redundancies, that are available for us to exploit.

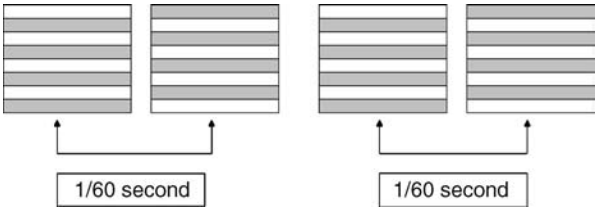


FIGURE 1.5 Interlaced formats.

1.4.1 Intrinsic Redundancy of Video Source

In the literature, video compression can be achieved with the removal of the four types of redundancy, spectral, temporal, spatial, and statistical redundancies. As shown in Figure 1.6, we list each category with approaches that can be used to remove such redundancy.

Spectral redundancy refers to the redundancy between each color component. Spectral redundancy can be removed with various color conversions, such as YUV, YIQ, and HSI. The majority of the information is compacted into the luminance components, and the chrominance components contain less visually sensitive information. Typically, the chrominance components can be subsampled horizontally and/or vertically to reduce the number of pixels without significantly degrading the visual quality. Thus, we can achieve about a factor of 2 compression ratio.

Temporal redundancy refers to the similarities between consecutive frames. Temporal redundancy can be removed with motion compensation (MC) or motion compensated temporal filtering (MCTF). The MCTF is used for a three-dimensional

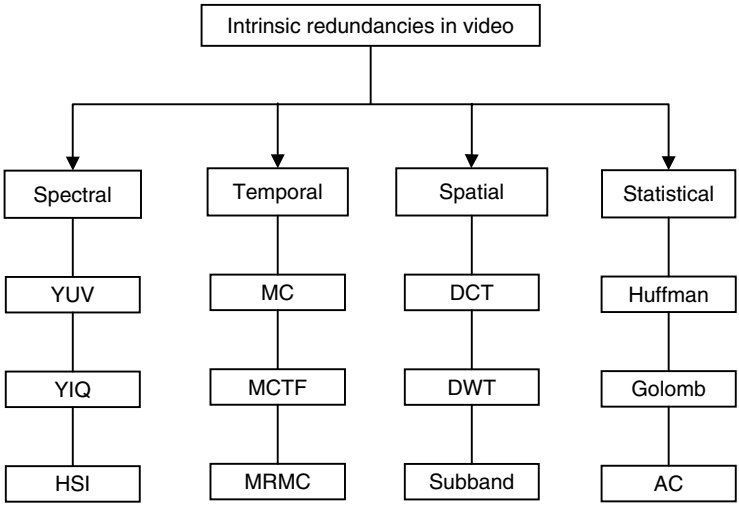


FIGURE 1.6 Intrinsic redundancies of video source.

(3-D) wavelet-based approach [1-25]. In the case of 3-D wavelet compression, the frames buffer can store $2n$ frames for improved compression. The benefits of increased complexity are scalability and coding efficiency. In the H.264 standard, multiple frames are allowed to be stored for predictions. This technique is referred to as *multiple reference frame motion compensation* (MRMC) [1-17].

Spatial redundancy refers to the correlation existent between spatially neighboring pixels. Spatial redundancy can be removed with various transforms, such as the discrete cosine transform (DCT), discrete wavelet transform (DWT), or subband decomposition. The discrete cosine transform provides very low cost implementations for energy compaction. There are fast algorithms and integer form for DCT implementation. The discrete cosine transform has been widely used in both image and video coding standards. Another tool for energy compaction is DWT. The discrete wavelet transform is mostly used for image compression and has recently been considered for video. Subband decomposition was widely used for audio coding and was used for image coding, which facilitates adaptation based on perceptual models.

Statistical redundancy is closely tied with the transform coding, in which most energy has been compacted toward the low-frequency components. The high-frequency components are more likely to become zero, and the nonzero coefficients are more likely to be located in the low-frequency area. Furthermore, the number of zero or close-to-zero coefficients is significantly larger than the number of coefficients with large amplitude. Thus, there is inherent statistical redundancy in the source that can be removed with Huffman, Golomb, or arithmetic code.

1.4.2 Temporal Redundancy

The second source of redundancy is the temporal redundancy coming from the temporal correlations because the same objects exist between frames when the sampling period is small enough such that no significant deformation occurs. Thus, a translation or more sophisticated model can describe the motion defined as a two-dimensional vector in the horizontal and vertical directions. The motion displacements are typically referred to as *motion vectors*. With the assistance of motion vectors, a block in the previous reconstructed frame is subtracted from the block in the current frame and the residual is encoded and transmitted to the decoder for reconstruction. The residual has significantly less information, which can approximately result in a further compression ratio of a factor 10.

There are several parameters that can be optimized based on the concept of motion vectors. We can optimize the performance according to the shape and size of the block, the motion model, the numerical precision of the vector, and the direction of the predictions. We will briefly describe the concepts behind these tools.

The motion models can be optimized based on various models, such as translation, affine, or perspective motion models. For the affine motion model, there are six parameters necessary to represent the motion. The overhead for transmitting the model parameters should be justified by the energy reduction of the residuals signal. The computation of the parameters is an important factor to consider when building the encoders.

The shape and size of the block can be optimized according to the physical object being predicted. The considered shapes are square, rectangle, triangle, hexagon, and

polygon. In most video coding standards, the first three shapes are often used. For the rectangular block, the size can vary from 4×4 to 16×16 or larger. In the MPEG-1 video standard, only the block size of 16×16 is used. The 16×8 block size was included in the MPEG-2 video, and the 8×8 block size and the triangular mesh were included in the MPEG-4 video specifications. Smaller block sizes, such as 4×4, 4×8, 8×4, have been added in the recently unveiled H.264. The trend is obvious that smaller block sizes are used to provide a more accurate description of the model, which can lead to better coding efficiency. However, it also implies increased complexity.

The numerical precision of the motion vectors is also critical. Half-pixel precision is used in MPEG-1 and MPEG-2, while quarter-pixel precision was used in MPEG-4 and H.264 for improved performance. The increased precision of the motion vectors has several implications. The motion is more accurately described so that the coding efficiency is improved. However, the bit overhead for sending accurate motion has to outweigh the bit savings from the reduced residual energy. Furthermore, fractional pixel resolution requires more computation for searching the motion parameters. For example, the search points for quarter pixel motion vectors are four times more than the half pixel motion vectors.

The direction and number of frames stored for motion prediction are critical for its performance. As shown in Figure 1.7, the current frame can be predicted from $t - 2$, $t - 1$, and $t + 1$. The use of the frame $t + 1$ is very important because it offers information about uncovered background or new objects that cannot be found from the past frames. Frame $t - 2$ offers more information at the cost of increased frame storage. Another variation of the multiple reference frames is to perform an average of the two predictions from two frames. This is typically referred to as *bidirectional* prediction. An average of two predictions was used in the MPEG standards and that has been proven to be very effective. One of the reasons for its effectiveness is that it is equivalent to fractional pixel motion compensation for translational motion. The bidirectional predicted frame is not available for future predictions in the MPEG-1/2/4 standards. Such a constraint is removed to achieve improved coding efficiency in the H.264 standards although it requires further delays and frame stores.

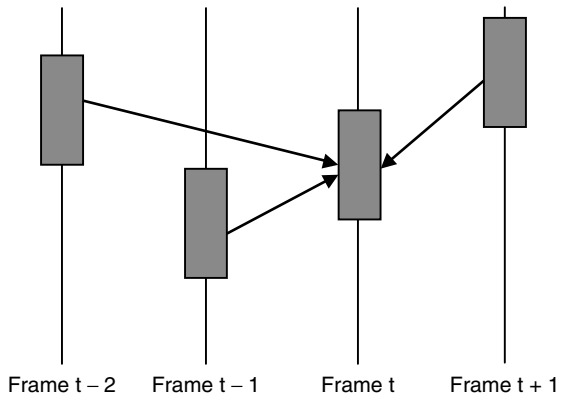


FIGURE 1.7 Motion compensation.

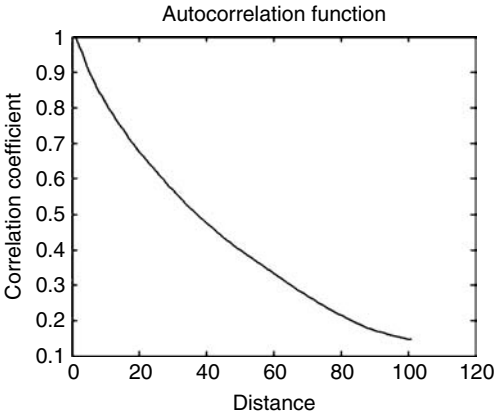


FIGURE 1.8 Autocorrelation functions of the Lena image.

1.4.3 Spatial Redundancy

The third source of redundancy is the spatial redundancy that comes from the correlations between spatially neighboring pixels. As shown in Figure 1.8, the spatial redundancy of a typical sequence can be approximated as a Markov model with using high correlations.

Typically, a compression ratio of 5 can be achieved with transformation using spatial redundancy removal. The transformations can be Discrete Cosine Transform (DCT), Discrete Since Transform (DST), and Discrete Wavelet Transform (DWT), or subband decomposition. The transformation takes advantage of the statistical redundancy to compact the energy into the low-frequency domain. The efficiency of energy compaction can be measured by the percentage of energy contained in the first few coefficients. As shown in Figure 1.9, the DCT coefficients have a Laplacian distribution that can be compressed with entropy coders easily.

1.5 BLOCK-BASED TRANSFORM

For video and image, there is an inherent spatial redundancy between neighboring pixels. To exploit such redundancy, a commonly used approach is to transform the spatial domain information into the frequency domain. There were several different transforms developed in the past. In the section, we will focus on the block-based transform coding and the next section will address the recently developed frame-based transform.

The theoretical optimal solution is the Karhunen-Loève transform, which that can achieve the most energy compaction. However, the evaluation of the Karhunen-Loève transform basis is input dependent. Thus, a less-sophisticated alternative of the transform has been sought. In this section, we will discuss various approaches to reach such a goal. In particular, we will study the Karhunen-Loève transform, discrete cosine transform, integer discrete cosine transform, subband decomposition and discrete wavelet transform.

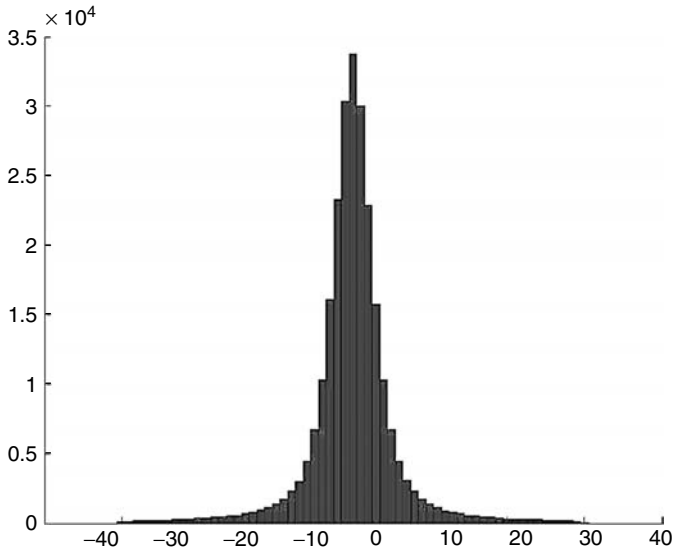


FIGURE 1.9 Histogram of DCT coefficients.

1.5.1 Karhunen-Loève transform

To fully exploit spatial redundancy, it is known that the Karhunen-Loève transform can provide the optimal solution. The rows of the discrete Karhunen-Loève transform consist of the eigenvectors for the autocorrelations matrix. The autocorrelation matrix for a stationary discrete random process $X(N)$ is given by

$$R = \begin{bmatrix} E(X_1^2) & E(X_1X_2) & \cdots & E(X_1X_n) \\ & E(X_2^2) & & \\ & & \ddots & \\ E(X_nX_1) & E(X_nX_2) & \cdots & E(X_n^2) \end{bmatrix}$$

$$= \begin{bmatrix} R_X(0) & R_X(1) & \cdots & R_X(N-1) \\ & R_X(0) & & \\ & & \ddots & \\ R_X(N-1) & R_X(N-2) & \cdots & R_X(0) \end{bmatrix}$$

It can be shown that this transform will minimize the geometric mean of the variance of the transform coefficients. Thus, it provides the best transform coding gain. However, the disadvantage of such an approach is that the source signals are typically nonstationary, so the autocorrelation function changes as time varies. Thus, the autocorrelation matrix needs to be recomputed for different information sources. Because the transform coefficient varies with time, it is necessary to transmit the coefficients to the decoder for correct decoding. The overhead or side information may not justify the additional coding gain it can achieve.

1.5.2 Discrete Cosine Transform

The discrete cosine transform (DCT) consists of a sum of cosine functions with different frequencies. The DCT is widely used in various standards, including JPEG, MPEG, and H.264. The DCT can be derived from the discrete Fourier transform (DFT), but the DCT is superior in several aspects. First, the computation of DCT does not involve any computation of complex numbers. Since it is derived from DFT, there are several known fast algorithms. Second, the DCT can perform better energy compaction for most correlated signals, such as images and video. It can be shown that the performance of DCT can approach that of KLT when we compress a Markov information source with high correlation coefficient ρ. Thus, DCT becomes the first choice for many compression standards.

The forward discrete cosine transform and inverse discrete cosine transform are defined as follows.

FDCT:

$$S_{uv} = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} s_{ij} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16}$$

IDCT:

$$s_{ij} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v S_{uv} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16}$$

$$C_u C_v = \begin{cases} \frac{1}{\sqrt{2}} & for\, u, v = 0 \\ 1 & otherwise \end{cases}$$

1.5.3 Fast Discrete Cosine Transform Algorithms

There are fast algorithms for implementing the DCT. In the early days, the DCT can be implemented with a double-size fast Fourier transform (FFT) algorithm using complex arithmetic [1-20]. In [1-18], Chen et al. describe an efficient algorithm using only real operations for computing the DCT of a set of N points, where $N = 2^n$ and $n > 1$. It consists of alternating cosine and sine butterfly pattern with matrices to reorder the matrix elements such that the bit-reversed pattern is preserved. The Chen algorithm takes $(3N/2)(\log_2 N - 1) + 2$ real additions and $N \log_2 N$ E $3N/2 + 4$ real multiplications. This algorithm offers six times faster speed compared to the conventional approach using double-size FFT [1-18].

1.5.4 Integer Discrete Cosine Transform

In the MPEG-1/2/4 standards, the residual signal after motion compensation is compressed with 8×8 floating-point precision DCT transform. For practical applications the floating-point transform is implemented with a finite precision. There is a mismatch

in the computation of DCT and IDCT at both the encoder and decoder. For static scenes, the errors in the IDCT mismatch will accumulate and result in visible artifacts. The mismatch can be removed by periodically inserting intra-coded block to stop the accumulation.

To solve the IDCT mismatch problem, a fundamental solution is to define the transform in such a way that finite precision implementation is possible. Thus, an integer DCT was defined in the recently developed H.264 standard [1-33]. There two different types of DCT for block size of 2×2 and 4×4. The forward integer DCT transform is defined as

$$\mathrm{DCT}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

and the inverse integer DCT transform is defined as

$$\mathrm{IDCT}_4 = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}$$

It is obvious that such a transform can be implemented with only additions and shift without any multiplications. In the H.264/AVC standards, the quantization process is jointly optimized so that 16-bit arithmetic is possible. Using very short tables, the quantization process can be implemented without any divisions [1-16].

1.5.5 Adaptive Block Size Transform

Similar to variable block size motion compensation, the transformation can be adaptive to the local statistics with variable block size transformation. The larger block size can provide better energy compaction and detail preservation for pictures with flat area and high spatial correlation. A larger block size can represent a large area with only a few coefficients. However, the smaller block size can address picture area with high details since the picture has higher spatial correlation when the block size is reduced. It also removes the ringing artifacts that typically appear in the larger transform or discrete wavelet transform. A flexible variable block transform also enables efficient intra-picture prediction and rate-distortion optimization.

The implementation of variable block size transformation can be formulated as

$$A = T_v \cdot B \cdot T_h^T$$

The matrices A and B represent the frequency and spatial blocks of the transform. In [1-19], a combination of block sizes of 8 and 4 are proposed to achieve variable

block size transformation. The block size-4 transformation matrix T_4 adopts the same transform matrix as specified in the H.264/AVC standard. The block size-8 transformation matrix T_8 can use the following matrix

$$\begin{pmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{pmatrix}$$

Such a size-8 transform can be implemented with an efficient fast algorithm using 36 additions, eight bit-shift operations, and ten multiplications as described in [1-19]. The improvement is about 12% in rate savings and 0.9 dB in peak signal-to-noise ratio.

1.6 FRAME-BASED TRANSFORM

In this section, several frame-based transforms are discussed. In particular, the subband and wavelet decompositions for image coding are studied. Using the frame-based wavelet transform, several image coding techniques exploit the parent-child relationships and the self-similarity property of the frequency bands at the same spatial location. In particular, we will study the well-known embedded zerotree wavelet (EZW) and set partitioning in hierarchical trees (SPIHT). Finally, generalization to the temporal directions for video coding are considered. In particular, we consider the 3-D wavelet and its variations.

1.6.1 Subband Decomposition

The concept of subband coding (SBC) is to partition the original image signal into multiple frequency bands using lowpass and highpass filters as shown in Figure 1.10. The first set of filters H_0 and H_1 are referred to as the *analysis filters* while the second set of filters G_0 and G_1 are referred to as the synthesis filters. Each frequency band is separately quantized and encoded. The advantage of SBC is that the statistical

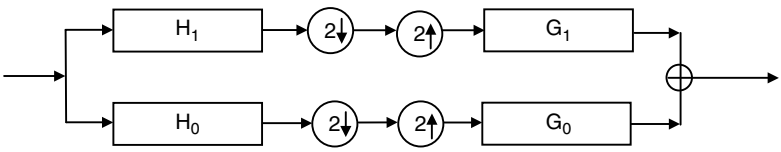


FIGURE 1.10 Filter bank for two channels.

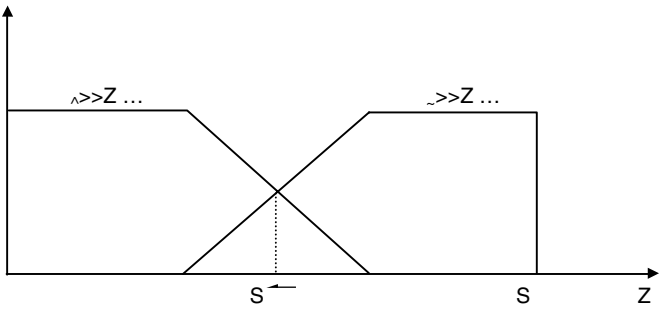


FIGURE 1.11 Quadrature mirror filters.

characteristics and redundancy of each band can be separately exploited with different quantization schemes and entropy coders. The bits can be dynamically allocated among different frequency bands and the distortion can be distributed according to human visual sensitivity.

In [1-23], Woods et al. studied the subband coding of images using quadrature mirror filters (QMF). As shown in Figure 1.11, the QMF frequency response for the one-dimension case is summed to be a constant so that perfect reconstruction at the receiver is possible. Woods et al. derived the constraints for a set of two-dimensional QMFs given a particular frequency partition. Furthermore, they demonstrate that separable one-dimensional QMFs can satisfy such constraints. There are many ways to partition the subbands. Typically, subband decomposition is performed with applications of lowpass and highpass filters in the horizontal and vertical directions of each frame. Each subband can be further decomposed into lowpass and highpass signals as illustrated in Figure 1.12. Another way to analyze the signal is to decompose the signal logarithmically. Since the human visual system is more sensitive to the distortions at the low-frequency components, it should be further analyzed with more resolution so that the encoder can perform the adaptation. The lowpass signals can be further analyzed to extract the specific frequency band as shown in Figure 1.12.

1.6.2 Discrete Wavelet Transform

In signal processing, the Fourier transform is commonly used to analyze signals in the frequency domain. The introduction of the discrete-time Fourier transform and its fast algorithm further expand the application of such an analytical tool. Fourier transform analysis can be considered as a linear expansion of a series of continuous bases made of *sine* and cosine functions. The Fourier series analysis can be also considered as using the set of periodic signals such as sines and cosines with infinite discrete frequencies as bases. The Fourier transform offers excellent resolution in the frequency domain but requires infinite time or buffer to compute because the bases expand from $-\infty$ to ∞. Such a phenomenon can be interpreted as indicating that the Fourier transform does not provide sufficient resolution in the time domain.

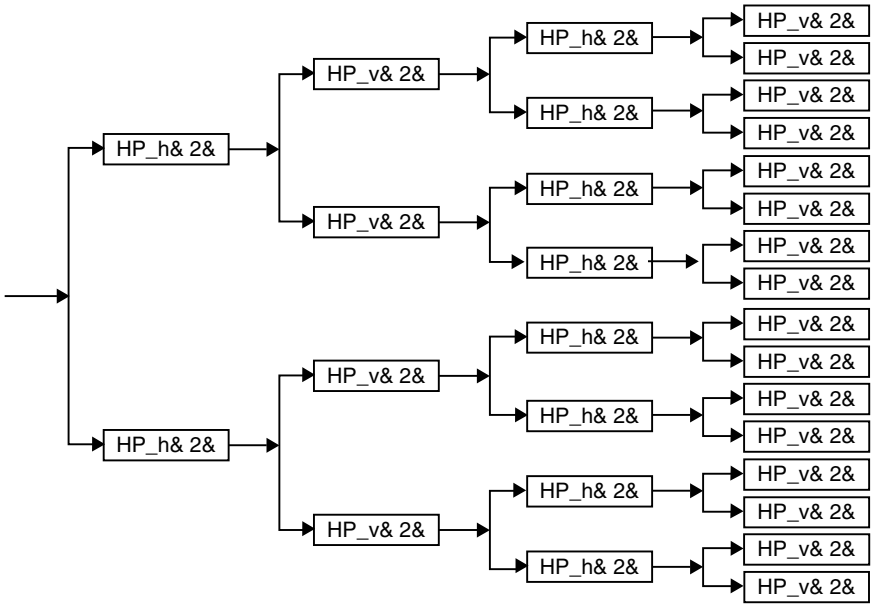


FIGURE 1.12 Equally partitioned subbands of a 2-D image.

The goal of the wavelet transform is to generalize the linear expansion concept and seek the expansion bases that can provide adequate tradeoff between time and frequency resolutions depending on the signal characteristics. The wavelet transform can use either orthonormal or biorthogonal bases or filters. The orthonormal filters conserve energy but lack the linear phase property. On the other hand, the biorthogonal filters can achieve linear phase as well as smoother approximation but cannot preserve energy, which makes it inconvenient for quantization and bit allocation.

The concept of better resolutions in the spatial or the frequency domain can be quantitatively measured as the space-frequency localization. With a measure for the spread in space and frequency of a function, one can define a region in the space-frequency plane where energy is mostly located. The wavelet filters can provide particular space-frequency localization regions or tiling with structured bases for expansion. The wavelet filters are related to each other through shifting, scaling, and/or modulations. For example, the short-time Fourier transform provides a rectangular tiling while the wavelet transform typically yields a dyadic tiling as shown in Figure 1.13 for the case of two-level analysis in the horizontal and vertical directions. The best tiling or bases is signal dependent. For example, a flat area in the picture or a lowpass signal may need better frequency resolution while a busy area in the picture may require better spatial resolution. The wavelet transform provides flexibility and tools to make such adaptations [1-29].

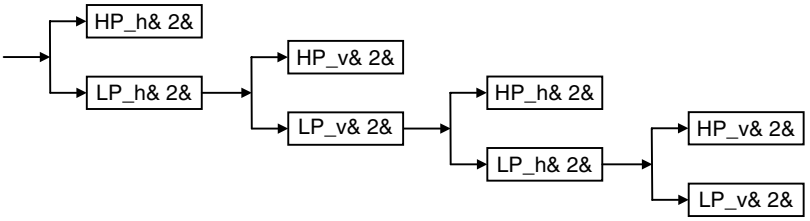


FIGURE 1.13 Two-level logarithmical partitioning of a 2-D image using the DWT.

Based on a single wavelet prototype function $\psi(t)$ with only shifting and scaling, one can construct a family of orthonormal bases $\psi_{mn}(t)$ as follows.

$$\psi_{mn}(t) = 2^{-mn}\psi\left(\frac{t-2^n}{2^m}\right)$$

In the case of dyadic tiling, the low frequency is repeatedly analyzed with the same discrete-time wavelet filter and so we are concerned whether the iterated filter is regular, i.e., such a filter converges to a continuous and differentiable function $\phi(t)$. Daubechies shows that the sufficient conditions for regularity can be obtained by placing a maximum number of zeros at $\omega = \pi$ and maintaining the orthogonality condition. Daubechies also provides a construction method to achieve regularity [1-30]. In Figure 1.14, the resultant signal is shown in Figure 1.15 with three levels of wavelet analysis.

As shown in Figure 1.14, the number of samples is identical with the original image and there are several inherent multiple spatial resolutions available in the decomposition. The multiresolution property of the wavelet transform is very desirable for progressive or error-resilient transmission.

For progressive transmission, the coarse low-frequency information can be sent first and the detail high-frequency information can be retrieved on request using additional bandwidth. The new information provides enhancement of the signal fidelity. As for the spatial resolution, the receiver can provide multiple spatial resolutions by extracting a subset of the wavelet subbands. In Figure 1.14, one can extract full, $^1/_4$, and $^1/_{16}$ resolutions of the original signal. For error-resilient transmission, the low-frequency information should be better protected depending on the transport scheme.

In a prioritized transmission scheme, the low frequency can be assigned to a channel with higher priority of less bit error rate or packet loss rate. In case of transmission errors, the receiver can still reconstruct a usable picture with less quality. This is a desirable feature, called *graceful degradation*, that is important for transmission over the Internet or terrestrial broadcasting.

1.6.3 Embedded Zerotree Wavelet

Shapiro pioneered the embedded zerotree wavelet (EZW) algorithm, which provides both progressive encoding of wavelet coefficients and excellent coding efficiency [1-26]. The wavelet coefficients are ordered according to the importance of the