VIDEO PROCESSING AND COMMUNICATIONS

Yao Wang
Jörn Ostermann
Ya-Qin Zhang

PRENTICE HALL
SIGNAL PROCESSING SERIES
ALAN V. OPPENHEIM, SERIES EDITOR

ELEMENT-1014
IPR2025-01108, IPR2025-01109

VIDEO PROCESSING AND COMMUNICATIONS

PRENTICE HALL SIGNAL PROCESSING SERIES
Alan V. Oppenheim, *Series Editor*

BRACEWELL *Two Dimensional Imaging*
BRIGHAM *The Fast Fourier Transform and Its Applications (AOD)*
BUCK, DANIEL, & SINGER *Computer Explorations in Signals and Systems Using MATLAB, 2/E*
CASTLEMAN *Digital Image Processing*
COHEN *Time-Frequency Analysis*
CROCHIERE & RABINER *Multirate Digital Signal Processing (AOD)*
JOHNSON & DUDGEON *Array Signal Processing (AOD)*
KAY *Fundamentals of Statistical Signal Processing, Vols. I & II*
KAY *Modern Spectral Estimation (AOD)*
LIM *Two-Dimensional Signal and Image Processing*
MCCLELLAN *et al. Computer-Based Exercises in Signal Processing using MATLAB5*
MENDEL *Lessons in Estimation Theory for Signal Processing, Communications, and Control, 2/e*
NIKIAS & PETROPULU *Higher Order Spectra Analysis*
OPPENHEIM & SCHAFER *Digital Signal Processing*
OPPENHEIM & SCHAFER *Discrete-Time Signal Processing*
OPPENHEIM & WILLSKY, with NAWAB *Signals and Systems, 2/e*
ORFANIDIS *Introduction to Signal Processing*
PHILLIPS & NAGLE *Digital Control Systems Analysis and Design, 3/e*
QUATIERI *Discrete-Time Speech Signal Processing: Principles and Practice*
RABINER & JUANG *Fundamentals of Speech Recognition*
RABINER & SCHAFER *Digital Processing of Speech Signals*
STEARNS & DAVID *Signal Processing Algorithms in MATLAB*
TEKALP *Digital Video Processing*
VAIDYANATHAN *Multirate Systems and Filter Banks*
VETTERLI & KOVACEVIC *Wavelets and Subband Coding*
WANG, OSTERMANN, & ZHANG *Video Processing and Communications*
WIDROW & STEARNS *Adaptive Signal Processing*

VIDEO PROCESSING AND COMMUNICATIONS

Yao Wang

Jörn Ostermann

Ya-Qin Zhang

Prentice Hall

PRENTICE HALL

Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data

Wang, Yao, 1962–
Video processing and communications / Yao Wang, Jörn Ostermann, Ya-Qin Zhang.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-017547-1
1. Image transmission. 2. Digital video I. Ostermann, Jörn. II. Zhang, Ya-Qin, 1969–
III. Title.

TK5105.2 .W36 2001
621.388′33—dc21 200103699

Vice President and Editorial Director, ECS: *Marcia J. Horton*
Publisher: *Tom Robbins*
Editorial Assistant: *Jessica Romeo*
Vice President and Director of Production and Manufacturing, ESM: *David W. Riccardi*
Executive Managing Editor: *Vince O'Brien*
Managing Editor: *David A. George*
Production Editor: *Kathy Ewing*
Director of Creative Services: *Paul Belfanti*
Creative Director: *Carole Anson*
Art Director: *Jayne Conte*
Art Editor: *Greg Dulles*
Manufacturing Manager: *Trudy Pisciotti*
Manufacturing Buyer: *Lynda Castillo*
Marketing Manager: *Holly Stark*
Marketing Assistant: *Karen Moon*

© 2002 by Prentice-Hall
Prentice-Hall, Inc.
Upper Saddle River, New Jersey

All rights reserved. No part of this book may be reproduced in any form
or by any means, without permission in writing from the publisher.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

MATLAB is a registered trademark of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 07160-2098.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN 0-13-017547-1

Pearson Education Ltd., *London*
Pearson Education Australia Pty. Ltd., *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Inc., *Toronto*
Pearson Educacíon de Mexico, S.A. de C.V.
Pearson Education—Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.

To Our Beloved Ones:

Hai and Yana
Dorota, Olivia, Oktavia, and Orry
Jenny, Sophie, and Brandon

VIDEO PROCESSING AND COMMUNICATIONS

Contents

PREFACE xxi

GLOSSARY OF NOTATIONS xxv

1 VIDEO FORMATION, PERCEPTION, AND REPRESENTATION 1

1.1 Color Perception and Specification 2
1.1.1 Light and Color, 2
1.1.2 Human Perception of Color, 3
1.1.3 The Trichromatic Theory of Color Mixture, 4
1.1.4 Color Specification by Tristimulus Values, 5
1.1.5 Color Specification by Luminance and Chrominance Attributes, 6

1.2 Video Capture and Display 7
1.2.1 Principles of Color Video Imaging, 7
1.2.2 Video Cameras, 8
1.2.3 Video Display, 10
1.2.4 Composite versus Component Video, 11
1.2.5 Gamma Correction, 11

1.3 Analog Video Raster 12
1.3.1 Progressive and Interlaced Scan, 12
1.3.2 Characterization of a Video Raster, 14

ery. *Numerical*
ity Press, 1992.
Detection. New

t-based coding.
5–79.
ee-dimensional
Trans. Pattern

parameters of a
gnal Processing

ee-dimensional
Transactions on
27.
ensional motion
on. *IEEE Trans-*
), 30(4):525–34.
two perspective
Transactions on
51–76.
Computer Vision

hrough partially
ion. *IEEE Trans-*
2), 14(1):19–34.

8

Foundations of Video Coding

Video coding is an important application of digital video processing. The goal of video coding is to reduce the data rate of a video sequence, so that it is feasible to transmit the video in real time over a given communication channel. The channel bandwidth varies, depending on the application and the transmission media. For video phone applications using regular telephone lines, 20 kbps may be all that is available for video coding. For broadcasting of standard-definition television signals over satellite, a data rate of 6 mbps may be available. In addition to communications applications, video coding is also necessary for storage and retrieval, where different storage media have different capacity and access rates, thus demanding varying amount of compression. Due to this wide range of data rates, different types of algorithms have been developed. One class of algorithms allows efficient coding of arbitrary video signals without trying to analyze the video content. A second class of algorithms identifies regions and objects in a video sequence and codes those. We call the former *waveform-based* video coders and the latter *content-dependent* video coders.

In this chapter, we first give an overview of different waveform-based and content-dependent video coders in Section 8.1. Then, following a review of basic probability and information theory concepts in Section 8.2, information theory used to optimize lossless coding, and rate distortion theory used to optimize lossy coding are introduced in Section 8.3. This is followed by descriptions of lossless coding (Section 8.4), and the most basic lossy coding techniques, including scalar and vector quantization (Sections 8.5 and 8.6). Chapters 9 and 10 will discuss techniques for waveform-based and content-dependent video coding, respectively.

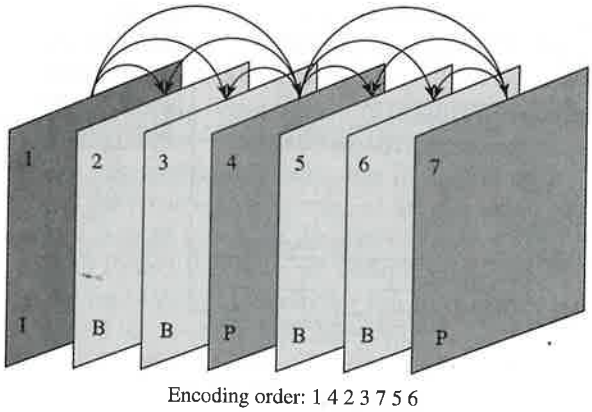

Figure 9.12 Video coding using both uni- and bidirectional temporal prediction. The arrows indicate the reference frames used for predicting a coded frame. Frames labeled I, P, and B are coded without prediction, with unidirectional prediction, and with bidirectional prediction, respectively.

bidirectional prediction (known as the PB-mode). On the other hand, the MPEG standard series, targeted mainly for video distribution, employs both uni- and bidirectional prediction.

9.3 VIDEO CODING USING TEMPORAL PREDICTION AND TRANSFORM CODING

A popular and effective video coding method is the one that uses block-based temporal prediction and transform coding. This method is essentially the core of all the international video coding standards. In this section, we first provide an overview of this coding method (Section 9.3.1). This will be followed by an extension, which uses more sophisticated motion-compensated prediction (Section 9.3.2). Next, we consider the issue of how to choose various coding parameters in this coding method to maximize the coding efficiency under the rate constraint imposed by the underlying application (Sections 9.3.3–4). As an optional advanced topic, we analyze the influence of motion estimation accuracy and loop filtering on coding efficiency (Section 9.3.5).

9.3.1 Block-Based Hybrid Video Coding

In this coder, each video frame is divided into blocks of a fixed size and each block is processed more or less independently, hence the designation "block-based." The word "hybrid" means that each block is coded using a combination of motion-compensated temporal prediction and transform coding. Figure 9.13 shows the key steps in this coding paradigm. First, a block is predicted from a previously coded reference frame using block-based motion estimation. The motion vector specifies the displacement between the current block and the best matching block. The predicted block is obtained from

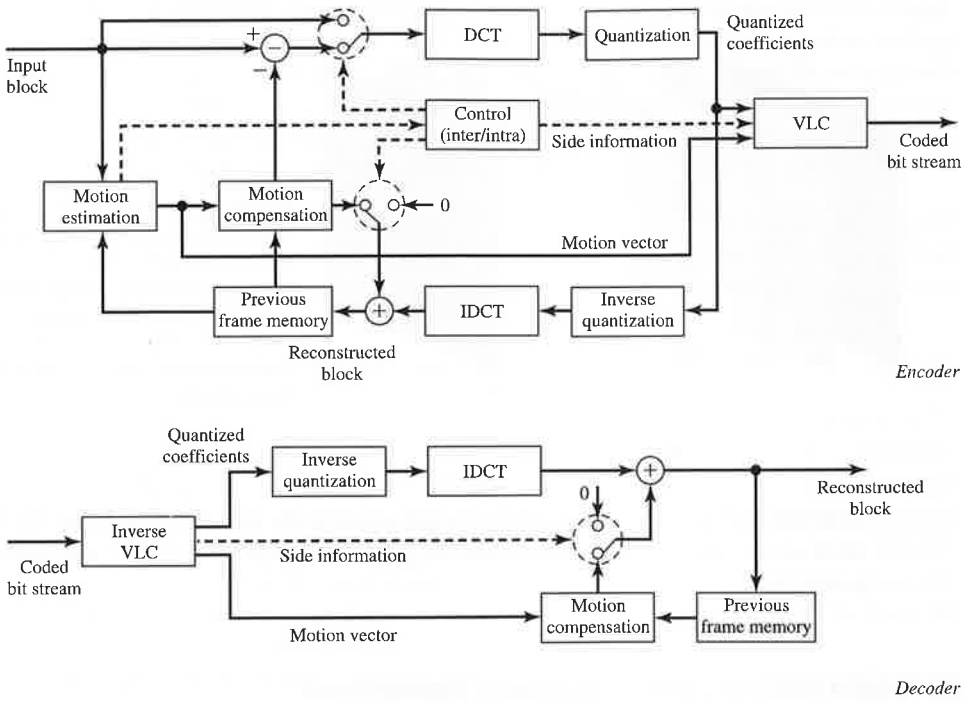

Figure 9.13 Encoding and decoding process for a block in a typical block-based hybrid coding system.

the previous frame based on the estimated MV using motion compensation.* Then, the prediction error block is coded, by transforming it using the DCT,† quantizing the DCT coefficients, and converting them into binary codewords using variable-length coding. The actual coding method is similar in principle to that described in Section 9.1.7. As with the JPEG standard, the quantization of the DCT coefficients is controlled by a quantization parameter, which scales a predefined quantization table. The quantization tables, quantizer implementations, and run length coding methods employed by various video coding standards are described in Chapter 13. Note that the encoder must emulate the decoder operation to deduce the same reconstructed frame as the decoder, so that there will be no mismatch between the reference frames used for prediction.

*The phrase "motion compensation," here and subsequently in this book, will refer to motion-compensated prediction.

†Ideally, if the prediction is very accurate, then the error pattern will be close to white noise and can be effectively coded using scalar quantization. In reality, however, the use of a block-based motion model often leads to structured error patterns. Transform coding is used to exploit the correlation between error pixels.

The preceding discussion assumes that temporal prediction is successful, in that the prediction error block requires fewer bits to code than the original image block. This method of coding is called *P-mode.* When this is not the case, the original block will be coded directly using transform coding. This is known as *intra-mode*. Instead of using a single reference frame for prediction, bidirectional prediction can be used, which finds two best matching blocks, one in a previous frame and another in a following frame, and uses a weighted average of the two matches as the prediction for the current block. In this case, two MVs are associated with each block. This is known as the *B-mode.* Both P- and B-modes are generally referred to as *inter-mode.* The mode information and the MVs, as well as other side information regarding picture format, block location, and so on, is also coded using VLC.

In practice, the block size for motion estimation may not be the same as that for transform coding. Typically, motion estimation is done on a larger block known as a *macroblock* (MB), which is subdivided into several blocks for which the DCT is evaluated. For example, in most video coding standards, the MB size is 16×16 pels and the block size is 8×8 pels. If the color subsampling format is 4:2:0, then each MB consists of 4 Y blocks, 1 Cb block, and 1 Cr block. The coding mode (i.e., intra- or inter-mode) is decided at the MB level. Because MVs and DC coefficients of adjacent MBs or blocks are usually similar, they are typically coded predictively, using the MV and DC coefficients of the previous MB or block as the predicted values. In all the video coding standards described in Chapter 13, a number of MBs form a *group of blocks* (GOB) or a *slice,* and several GOBs or slices form a picture. Sizes and shapes of GOBs and slices differ among the various video coding standards and picture sizes, and can often be tailored to the application's needs. Prediction of MVs and DC coefficients are usually restricted within the same GOB or slice. That is, the predicted MV or DC value for the first MB or block in a GOB is set to some default value. This is to suppress error propagation when the compressed bit stream is corrupted by transmission or storage errors.

A frame may be coded entirely in the intra-mode, and such a frame is called an *intra-frame* or I-frame. The first frame of a sequence is always coded as an intra-frame. In applications employing high bit rates, or with relaxed real-time constraints, intra-frames are also used periodically to stop potential error propagation, and to enable random access. (Low-latency applications, such as video conferencing, cannot use such periodic refresh, because the number of bits from an intra-frame is typically much greater than for a predicted frame, and thus will cause a sudden surge in the transmitted data.) A *P-frame* uses only a past frame for prediction, and depending on the prediction accuracy, an MB can be coded in either intra- or P-mode, which is usually determined based on the prediction error. Finally, a *B-frame* uses bidirectional prediction, and an MB in a B-frame can be coded in intra-, P-, or B-mode. A B-frame can only be coded after the surrounding I- or P-frames are coded (see Figure 9.12). Both P- and B-frames are generally referred as *inter-frames.*

In the MPEG-1 and MPEG-2 standards, the frames are partitioned into *groups of pictures* (GOPs), with each GOP starting with an I-frame, followed by interleaving P- and B-frames. (A typical GOP is illustrated in Figure 9.12). This enables random access: one can access any GOP without decoding previous GOPs. The GOP structure

also enables fast-forwarding and rewinding: fast-forwarding can be accomplished by decoding only I- or I- and P-frames. Fast rewinding can be realized by decoding only I-frames, in a backward order.

9.3.2 Overlapped Block Motion Compensation

In this and the next subsection, for notational simplicity, we do not specify the time index in the video function. Rather, we use $\psi(\mathbf{x})$, $\psi_r(\mathbf{x})$, and $\psi_p(\mathbf{x})$ to represent the coded frame, the reference frame, and the predicted frame, respectively. In block-based motion compensation, every pixel in a block is predicted by a single pixel in the reference frame, using

$$\psi_p(\mathbf{x}) = \psi_r(\mathbf{x} + \mathbf{d}_m), \quad \mathbf{x} \in \mathcal{B}_m, \tag{9.3.1}$$

where $\mathbf{d}_m$ represents the MV of the block $\mathcal{B}_m$. When $\mathbf{d}_m$ is noninteger, an interpolation is needed to obtain $\psi_r(\mathbf{x} + \mathbf{d}_m)$. As previously described, when the estimated MV is inaccurate, or when the blockwise translational model is not appropriate, the predicted image will suffer from the notorious blocking artifact. To overcome this problem, two approaches have been taken. One is to combine the motion estimates for neighboring blocks to produce the predicted value for a given pixel, and another is to apply filtering to the predicted image. We discuss the first approach in this section, and the filtering approach in Section 9.3.5.

With overlapped block motion compensation (OBMC) [25, 26], a pixel is predicted based not only on the estimated MV for the block it belongs to, but also the MVs of its neighboring blocks, as illustrated in Figure 9.14. Let K be the total number of neighboring blocks considered, and let $\mathcal{B}_{m,k}$ and $\mathbf{d}_{m,k}$, $k \in \mathcal{K} = \{0, 1, 2, \ldots, K\}$,

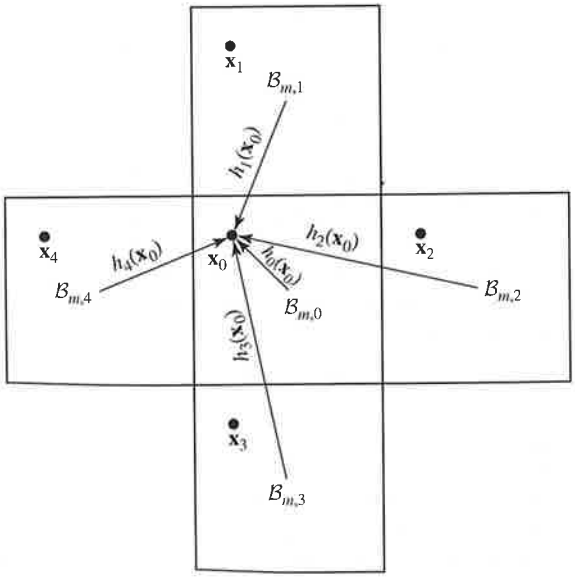

Figure 9.14 An example of overlapped block motion compensation, using four neighboring blocks.