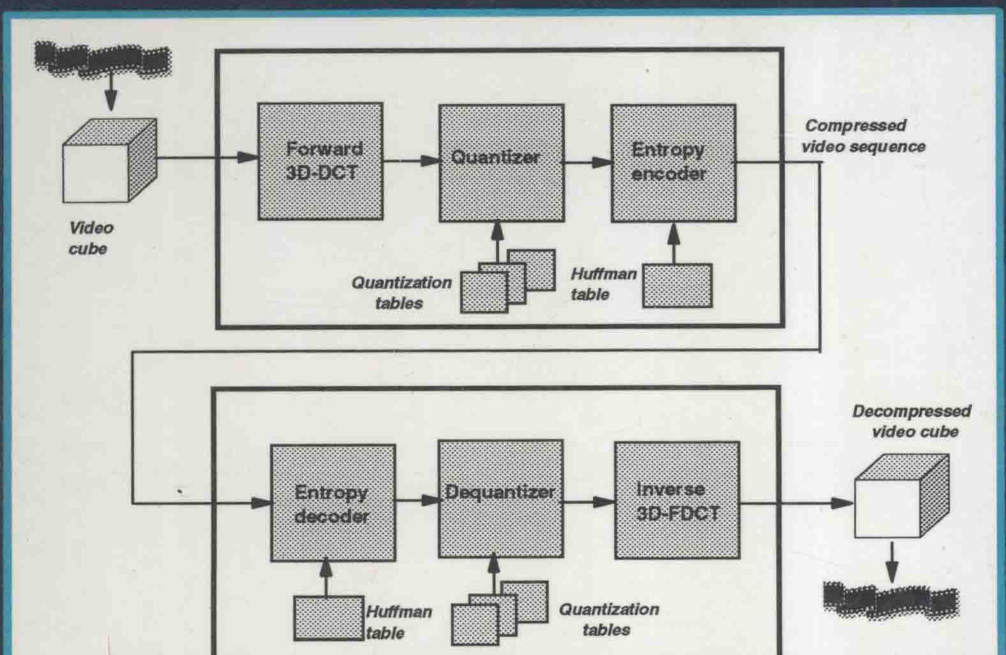
REAL-TIME VIDEO COMPRESSION

Techniques and Algorithms

Raymond Westwater
Borko Furht



REAL-TIME VIDEO COMPRESSION
Techniques and Algorithms

by

Raymond Westwater
Borko Furht
Florida Atlantic University

KLUWER ACADEMIC PUBLISHERS
Boston / Dordrecht / London

Distributors for North America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

Printed on acid-free paper.

Printed in the United States of America

REAL-TIME VIDEO COMPRESSION
Techniques and Algorithms

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

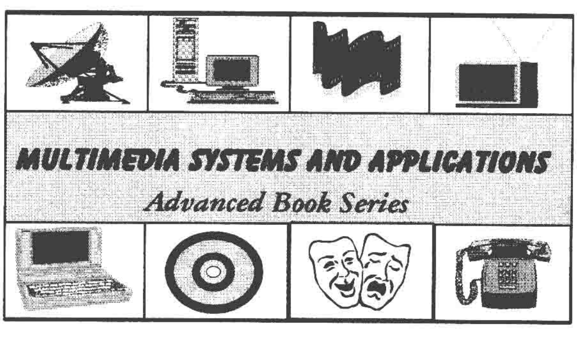
MULTIMEDIA SYSTEMS AND APPLICATIONS

Consulting Editor

Borko Furht
Florida Atlantic University

Recently Published Titles:

VIDEO AND IMAGE PROCESSING IN MULTIMEDIA SYSTEMS, by Borko Furht, Stephen W. Smoliar, HongJiang Zhang
ISBN: 0-7923-9604-9
MULTIMEDIA SYSTEMS AND TECHNIQUES, edited by Borko Furht
ISBN: 0-7923-9683-9
MULTIMEDIA TOOLS AND APPLICATIONS, edited by Borko Furht
ISBN: 0-7923-9721-5
MULTIMEDIA DATABASE MANAGEMENT SYSTEMS, by B. Prabhakaran
ISBN: 0-7923-9784-3



PREFACE

This book is on real-time video compression. Specifically, the book introduces the XYZ video compression technique, that operates in three dimensions, eliminating the overhead of motion estimation. First, video compression standards, MPEG and H.261/H.263, are described. They both use asymmetric compression algorithms, based on motion estimation. Their encoders are much more complex than decoders. The XYZ technique uses a symmetric algorithm, based on the Three-Dimensional Discrete Cosine Transform (3D-DCT). 3D-DCT was originally suggested for compression about twenty years ago, however at that time the computational complexity of the algorithm was to high, it required large buffer memory, and was not as effective as motion estimation. We have resurrected the 3D-DCT based video compression algorithm by developing several enhancements to the original algorithm. These enhancements made the algorithm feasible for real-time video compression in applications such as video-on-demand, interactive multimedia, and videoconferencing. The demonstrated results, presented in the book, suggest that the XYZ video compression technique is not only a fast algorithm, but also provides superior compression ratios and high quality of the video compared to existing standard techniques, such as MPEG and H.261/H.263. The elegance of the XYZ technique is in its simplicity, which leads to inexpensive VLSI implementation of a XYZ codec.

We would like to thank Jim Prince for conducting experiments in developing visually weighted quantizers for the XYZ algorithm, as well as a number of students from Florida Atlantic University, who participated in these experiments. We also want to thank Drs. Roy Levow, K. Genesan, and Matthew Evett, professors from Florida Atlantic University, Dr. Steve Rosenbaum from Cylex Systems, and Joshua Greenberg for constructive discussions during this project.

Raymond Westwater and Borko Furht
Boca Raton, July 1996.

REAL-TIME VIDEO COMPRESSION
Techniques and Algorithms

CONTENTS

1

THE PROBLEM OF VIDEO COMPRESSION

The problem of real-time video compression is a difficult and important one, and has inspired a great deal of research activity. This body of knowledge has been, to a substantial degree, embodied into the MPEG and H.261/H263 motion video standards. However, some important questions remain unexplored. This book describes one possible alternative to these standards that has superior compression characteristics while requiring far less computational power for its full implementation.

Since about 1989, moving digital video images have been integrated with programs. The difficulty in implementing moving digital video is the tremendous bandwidth required for the encoding of video data. For example, a quarter screen image (320 x 240 pixels) playing on an RGB video screen at full speed of 30 frames per second (fps) requires storage and transmission of 6.9 million bytes per second. This data rate is simply prohibitive, and so means of compressing digital video suitable for real-time playback are a necessary step for the widespread introduction of digital motion video applications.

Many digital video compression algorithms have been developed and implemented. The compression ratios of these algorithms varies according to the subjective acceptable level of error, the definition of the word compression, and who is making the claim. Table 1.1 summarizes video compression algorithms, their typical compression ratios reported in the literature, and their characteristics.

Compression Algorithm	Typical Compression Ratio	Characteristics
Intel RTV/Indeo	3:1	A 128X240 data stream is interpolated to 256X240. Color is subsampled 4:1. A simple 16 bit codebook is used without error correction. Frame differencing is used.
Intel PLV	12:1	A native 256X240 stream is encoded using vector quantization and motion compensation. Compression requires specialized equipment.
IBM Photomotion	3:1	An optimal 8-bit color palette is determined, and run-length encoding and frame differencing are used.
Motion JPEG	10:1	Uses 2-D DCT to encode individual frames. Gives good real-time results with inexpensive but special-purpose equipment. This technique supports random-access since no frame differencing is used.
Fractals	10:1	Fractals compress natural scenes well, but require tremendous computing power.
Wavelets	20:1	2-D and 3-D wavelets have been used in the compression of motion video. Wavelet compression is low enough in complexity to compress entire images, and therefore does not suffer from the boundary artifacts seen in DCT-based techniques.
H.261/H263	50:1	Real-time compression and decompression algorithm for video telecommunications. It is based on 2-D DCT with simple motion estimation between frames.
MPEG	30:1	Uses 2-D DCT with motion estimation and interpolation between frames. The MPEG standard is difficult and expensive to compress, but plays back in real-time with inexpensive equipment.

Table 1.1 Overview of video compression algorithms.

An ideal video compression technique should have the following characteristics:

- Will produce levels of compression rivaling MPEG without objectionable artifacts.
- Can be played back in real time with inexpensive hardware support.

- Can degrade easily under network overload or on a slow platform.
- Can be compressed in real time with inexpensive hardware support.

1.1 OVERVIEW OF VIDEO COMPRESSION TECHNIQUES

The JPEG still picture compression standard has been extremely successful, having been implemented on virtually all platforms. This standard is fairly simple to implement, is not computationally complex, and gets 10:1 to 15:1 compression ratios without significant visual artifacts. This standard is based upon entropy encoding of quantized coefficients of the discrete cosine transformation of 8x8 blocks of pixel data.

Figure 1.1 shows the block diagram of both the JPEG compression and decompression algorithms. A single frame is subdivided into 8x8 blocks, each of which is independently processed. Each block is transformed into DCT space, resulting in an 8x8 block of DCT coefficients. These coefficients are then quantized by integer division by constants. The quantizing constant for each DCT coefficient is chosen to produce minimal visual artifacts, while maximally reducing the representational entropy of the coefficients. The quantized coefficients are then entropy coded into a compressed data stream. The reduced entropy of the quantized coefficients is reflected in the higher compression ratio of the data.

The Motion JPEG (M-JPEG) uses the JPEG compression for each frame. It provides random access to individual frames, however the compression ratios are too low (same as in JPEG), because the technique does not take advantage of the similarities between adjacent frames.

The MPEG moving compression standard is an attempt to extend DCT-based compression into moving pictures. MPEG encodes frames by estimating the motion difference between the frames, and encoding the differences into roughly JPEG format. Unfortunately, motion estimation is computationally complex, requires specialized equipment to encode, and adds considerable complexity to the algorithm. Figure 1.2 illustrates the MPEG compression algorithm for predictive frames.

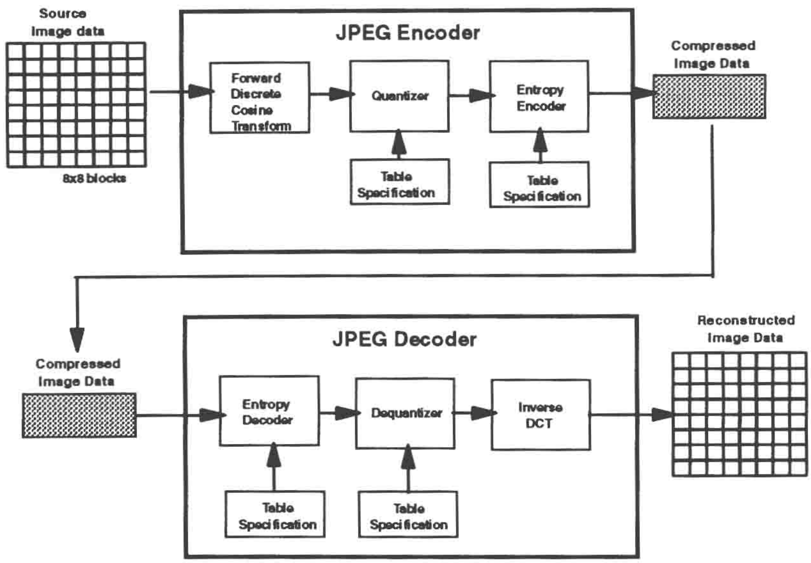


Figure 1.1 JPEG compression and decompression algorithms.

One of the most promising new technologies is wavelet-based compression [VK95]. Figure 1.3 illustrates a simple wavelet transform: subband decomposition. The image as a whole is subdivided into frequency subbands, which are then individually quantized. One of the most attractive features of this system is that it is applied to the image as a whole, thereby avoiding the edge artifacts associated with the block-based DCT compression schemes.

The wavelet transform can be applied to the time dimension as well. Experience has shown that this decomposition does not give as good compression results as motion compensation. As there are no other compression algorithms capable of such high compression ratios, MPEG is considered the existing "state-of-the-art".

The XYZ algorithm is a natural extension of the research that has been done in video compression. Much work has been done in the development of transform-based motion video compression algorithms, and in the development of quantizing factors based on the sensitivity of the human eye to various artifacts of compression.

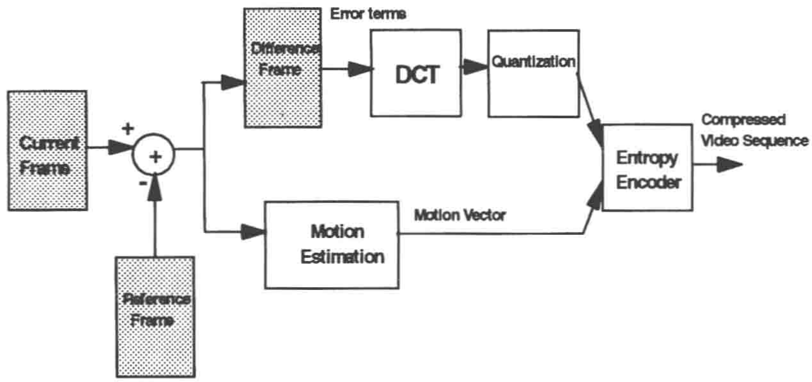


Figure 1.2 MPEG compression algorithm for predictive frames.
MPEG adds motion estimation to the JPEG model.

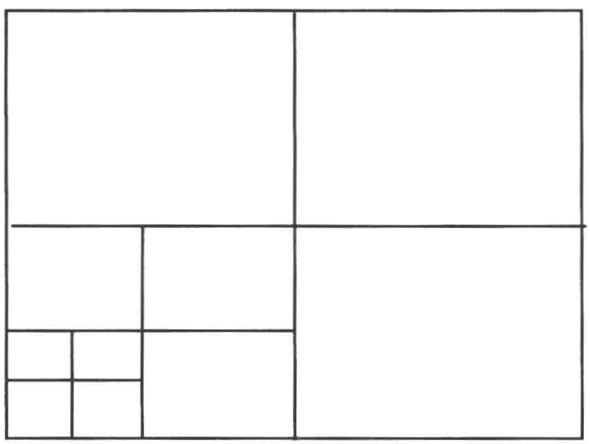


Figure 1.3 Octave-band or wavelet decomposition of a still image into unequal subbands.

XYZ compression is an alternative extension of DCT encoding to moving pictures. Sequences of eight frames are collected into a three-dimensional block to which a three-dimensional.DCT will be applied. The transformed data is then quantized.

These quantizing constants are demonstrated to cause artifacts which are minimally visible. The resulting data stream is then entropy coded. This process strongly resembles the JPEG encoding process, as illustrated in Figure 1.4.

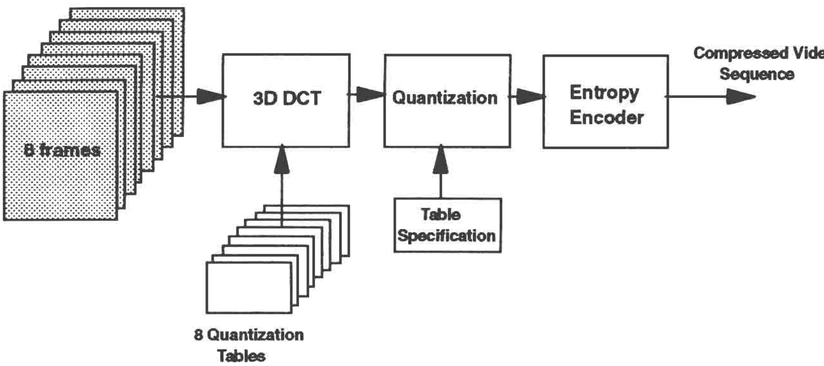


Figure 1.4 XYZ compression algorithm.

This algorithm is built upon a considerable body of published work. The three-dimensional DCT has been used to encode errors after motion estimation has been performed [RP77], and true three-dimensional DCT-based compression algorithms have been developed where the quantizers were based upon minimization of introduced mean square error [NA77]. These algorithms have fallen into disfavor because they were considered to require excessive computation, required too much buffer memory, and were not as effective as motion estimation. This book refutes these arguments.

Work in visibility of artifacts produced by quantization has also been done [CR90]. Visibility of two-dimensional quantization artifacts has been thoroughly explored for the DCT transforms space. The XYZ algorithm extends this work to quantization of three-dimensional DCT coefficients.

1.2 APPLICATIONS OF COMPRESSED VIDEO

Video compression techniques made feasible a number of applications. Four distinct applications of the compressed video can be summarized as: (a) consumer broadcast television, (b) consumer playback, (c) desktop video, and (d) videoconferencing.

Consumer broadcast television, which includes digital video delivery to homes, typically requires a small number of high-quality compressors and a large number of low-cost decompressors. Expected compression ratio is about 50:1.

Consumer playback applications, such as CD-ROM libraries and interactive games, also require a small number of compressors and a large number of low-cost decompressors. The required compression ratio is about 100:1.

Desktop video, which includes systems for authoring and editing video presentations, is a symmetrical application requiring the same number of encoders and decoders. The expected compression ratio is in the range from 5:1 to 50:1.

Videconferencing applications also require the same number of encoders and decoders, and the expected compression ratio is about 100:1.

Application	Bandwidth	Standard	Size	Frame Rate [frames/sec]
Analog Videophone	5-10 Kbps	none	170x128	2-5
Low Bitrate Video Conferencing	26-64 Kbps	H.263	128x96 176x144	15-30
Basic Video Telephony	64-128 Kbps	H.261	176x144 352x288	10-20
Video Conferencing	>= 384 Kbps	H.261	352x288	15-30
Interactive Multimedia	1-2 Mbps	MPEG-1	352x240	15-30
Digital TV - NTSC	3-10 Mbps	MPEG-2	720x480	30
High Definition Television	15-80 Mbps	MPEG-2	1200x800	30-60

Table 1.2 Applications of the compressed video and current video compression standards.

Table 1.2 summarizes applications of the compressed video, by specifying current standards used in various applications, the required bandwidth, and typical frame sizes and frame rates.

1.3 IMAGE AND VIDEO FORMATS

A digital image represents a two-dimensional array of samples, where each sample is called a pixel. Precision determines how many levels of intensity can be represented, and is expressed as the number of bits/sample. According to precision, images can be classified into: (a) binary images, represented by 1 bit/sample, (b) computer graphics, represented by 4 bits/sample, (c) grayscale images, represented by 8 bits/sample, and color images, represented with 16, 24 or more bits/sample.

According to the trichromatic theory, the sensation of color is produced by selectively exciting three classes of receptors in the eye. In a RGB color representation system, shown in Figure 1.5, a color is produced by adding three primary colors: red, green, and blue (RGB). The straight line, where R=G=B, specifies the gray values ranging from black to white.

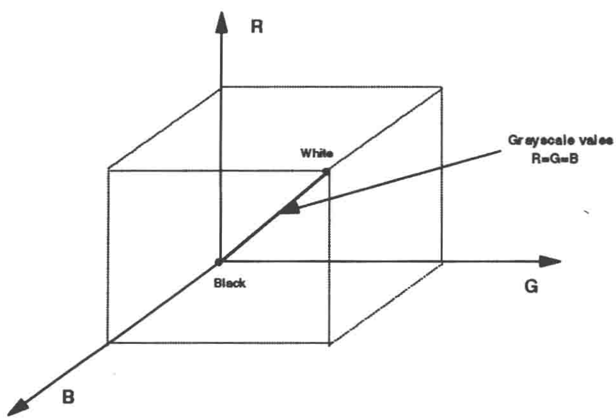


Figure 1.5 The RGB representation of color images.

Another representation of color images, YUV representation, describes luminance and chrominance components of an image. The luminance component provides a grayscale version of the image, while two chrominance components give additional information that converts the grayscale image to a color image. The YUV representation is more natural for image and video compression. The exact transformation from RGB to YUV representation, specified by the CCIR 601 standard, is given by the following equations: