

ICICS-PCM 2003

Nanyang
Technological University

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore



Proceedings of the

2003 Joint Conference of the
Fourth International Conference on Information,
Communications & Signal Processing
And
Fourth Pacific-Rim Conference on Multimedia

Volume I

15 - 18 December 2003

Meritus Mandarin Singapore Hotel, Singapore

TK5102.9

.I62

2003

v.1

quarto

**Proceedings of the 2003 Joint Conference of the Fourth International Conference
on Information, Communications and Signal Processing and Fourth Pacific-Rim
Conference on Multimedia (ICICS-PCM 2003)**

Copyright and Reprint Permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For other copying, reprint or republication permissions, write to IEEE Copyrights Manager, IEEE Operations Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. All rights reserved. Copyright ©2003 by the Institute of Electrical and Electronics Engineers, Inc.

IEEE Catalog Number: 03EX758

ISBN: 0-7803-8185-8

Library of Congress: 2003111734

6/16/2004 TM

Nanyang
Technological University

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore



Organised by

Technical Sponsorships by

IEEE Singapore Communications Chapter
IEEE Singapore Signal Processing Chapter



Sponsored by

Microsoft Research Asia

Microsoft

And

Singapore Exhibition and Convention Bureau



SINGAPORE EXHIBITION & CONVENTION BUREAU
A DIVISION OF THE SINGAPORE AIRPORT AUTHORITY

LINDA HALL LIBRARY
Kansas City, Mo.

ICICS-PCM 2003
2003 Joint Conference of the
Fourth International Conference on Information,
Communications & Signal Processing
And
Fourth Pacific-Rim Conference on Multimedia

Honorary Conference Chairs

M H Er, Singapore
R Gray, USA
S Y Kung, USA

Organising Committee

Advisors

S N Koh
A C Kot
H J Zhang

General Chairs

K H Li
P Xue

General Vice-Chairs

Z Lin
S F Chang

Program Chairs

S S Abeysekera
Q Tian

Program Vice-Chairs

A Alphones
T H Chiang

Secretary

S W Foo

Special Session

B Barkat
W Y Ma

Tutorial Chairs

V K Dubey
T S Ho

Finance Chairs

K C Teh
C Zhu

Exhibition/Sponsorship Chairs

Y L Guan
M Y Siyal

Local Arrangement Chairs

B P Ng
L W Yu

Publication Chairs

G X Xiao
K H Yap

Publicity Chairs

E Gunawan
C R Wan

Australia Liaison:

H R Wu

Hong Kong SAR Liaison:

W K Cham

Japan Liaison:

K Aizawa

Korea Liaison:

Y S Ho

USA Liaison:

T H Chen

International Program Committee

H Aghvami, UK
K Aizawa, Japan
B M Ali, Malaysia
J Astola, Finland
E Biglieri, Italy
A Cantoni, Australia
X Cao, Hong Kong SAR
T Constantinides, UK
W K Cham, Hong Kong SAR
K T Chan, Hong Kong SAR
K Chang, USA
W W Chang, Taiwan
C W Chen, USA
J J Y Chen, Taiwan
K Chen, UK
G M Chiu, Taiwan
T S Chua, Singapore
J H Chuang, Taiwan
C P Chung, Taiwan
I Claesson, Sweden
P Dini, USA
P M Djuric, USA
J L Dugelay, France
K Eshraghian, Australia
D J Evans, UK
S Furui, Japan
H Ghafouri-Shiraz, UK

Y H Gong, USA
H M Hang, Taiwan
L Hanzo, UK
Y S Ho, Korea
Y Hong, Hong Kong SAR
J N Hwang, USA
L C Jain, Australia
J Jin, Australia
M Kanchanalak, Singapore
S Katagiri, Japan
M Khalid, Malaysia
J C Kuo, USA
R L Langley, UK
C H Lee, USA
C S Li, USA
M Liao, Taiwan
X G Lin, China
T J Lim, Canada
A Mauthe, Germany
P Mulhem, France
T S Ng, Hong Kong SAR
T Ohira, Japan
F M B Pereira, Portugal
H V Poor, USA
R Prasad, Denmark
Y Sakai, Japan
K Salamatian, France
H Satoh, Japan
Y Q Shi, USA
T K Shih, Taiwan
U Srinivasan, Australia
D Slock, France
L Sornmo, Sweden
H F Sun, USA
Q B Sun, Singapore
S Valaee, Canada
B Vucetic, Australia
C N Wang, Taiwan
H R Wu, Australia
J K Wu, Singapore
L D Wu, China
C S Xu, Singapore
C Xydeas, UK
M M Yeung, USA
S Yoshida, Japan
A D Zhang, USA
Y J Zhang, China
B Zeng, Hong Kong SAR
C X Zheng, China
J M Zurada, USA



MESSAGE

General Chairs – ICICS-PCM 2003



On behalf of the Conference Organising Committee, we would like to welcome you to the 2003 Joint Conference of the Fourth International Conference on Information, Communications & Signal Processing (ICICS 2003) and the Fourth Pacific-Rim Conference on Multimedia (PCM 2003). The Committee has been working very hard during the last 12 months, especially during the SARS period, to ensure the success of this event. The Committee is very pleased to have received over 600 technical papers from 33 countries.

The joint conference promises to be a superb platform for individual engineers, companies and academia to report their recent developments and exchange their research ideas in the exciting areas of information engineering, communications, signal processing and multimedia. The invaluable interaction will allow us to stay on top of current technologies and meet the challenges ahead.

The Committee has invited two renowned keynote speakers, Professor Lawrence RABINER of Rutgers University and UC Santa Barbara, and Professor S Y KUNG of Princeton University. We are certain that you will enjoy their inspiring speeches.

The success of this conference would not have been possible without the support of many individuals and organizations. The joint conference is organized and sponsored by the School of Electrical and Electronic Engineering, Nanyang Technological University. Special thanks go to the Dean of the School, Professor Meng Hwa ER, for his guidance and encouragement throughout the whole event. We wish to thank the technical co-sponsorship of IEEE, especially IEEE Singapore Section, IEEE Singapore Communications Chapter and IEEE Singapore Signal Processing Chapter. The generous financial assistance from Microsoft and Singapore Exhibition & Convention Bureau is greatly appreciated.

It is a pleasure to thank the members of the Conference Organising Committee and the International Program Committee for their contributions. In particular, we are grateful to our Advisors, Professor Soo Ngee KOH and Professor Alex KOT from Nanyang Technological University and Dr. Hong-Jiang ZHANG from Microsoft Research Asia, for sharing their valuable ideas and experience with us. Last but not least, we would like to convey our appreciation to the keynote speakers, special session organizers, session chairs, authors and reviewers of technical papers, tutorial speakers and all the delegates for their contributions.

Singapore is one of Asia's most exciting and dynamic regions, where East blends harmoniously with West, tradition with modernity and sophistication with nature, creating a unique and stimulating urban experience. We trust that you will take back with you happy memories of Singapore and of this joint conference.

Kwok Hung LI

Ping XUE



MESSAGE

PROGRAM CHAIRS – ICICS-PCM 2003



On behalf of the Technical Program Committee, it is our pleasure and privilege to welcome you all to ICICS-PCM 2003. This is the first time that the ICICS and PCM are organised by the School of Electrical and Electronic Engineering, Nanyang Technological University as a Joint Conference. As a result the Technical Program Committee has organized an exciting technical program covering the areas of Communications, Signal Processing and Multimedia.

We have received a total of 623 papers from 33 countries, covering Asia, Australia, Europe and the Americas. Each paper was subjected to a thorough and comprehensive review by at least two independent reviewers. Based on the reviews we have selected 461 high quality papers for oral presentation, of which 46 papers appear in the special sessions. The technical papers have been arranged into 6 parallel sessions, covering the main technical areas of the conference - Multimedia Systems, Computer Systems, Signal Processing, Communications and Networking. The 8 special sessions of the technical program deal with topical and emerging areas of research.

Besides technical sessions, the conference also features 2 keynote speeches and 4 short courses/tutorials. In these, we are fortunate to have 5 well known authorities presenting in depth matter related to communications, signal processing and multi-media. We are extremely delighted to have two distinguished keynote addresses discussing issues on developments and trends in cutting-edge technologies. The first keynote address on Tuesday morning is "Telecom Technology for the 21st Century" by Professor L Rabiner (Rutgers University, U.S.A.), while the other keynote address "Data Mining on DNA Micro-array Data for Gene Selection and Classification" by Professor S Y Kung (Princeton University, U.S.A.), is on Wednesday morning.

Finally, we would like to thank the International Program Committee members as well as all our authors, reviewers, track-chairs, tutorial and special session organizers, for their hard work. Without their efforts this fine technical program would not have been a reality. We would also like to express our deep appreciation to our keynote speakers for giving us their valuable time.

We look forward to your participation in ICICS-PCM 2003 and hope you will enjoy your visit to Singapore and take with you some fond memories and new ideas to pursue.

Saman S ABEYSEKERA

Qi TIAN

Contents

Volume I

1A1 - Home Media Albums

- 1A1.1 P0911** 1
An Information-Integration Approach to Designing Digital Video Albums
*C Madhwacharyula, *P Mulhem, M Kankanhalli*
National University of Singapore, SINGAPORE
*IPAL-CNRS, FRANCE
- 1A1.2 P0912** 6
Photoc: Automatic Clustering for Browsing Personal Photographs
J Platt, M Czerwinski, B Field
Microsoft Research, UNITED STATES
- 1A1.3 P0909** 11
Personalized Karaoke
X S Hua, L Lu, H J Zhang
Microsoft Research Asia, CHINA
- 1A1.4 P0910** 16
Detection of Non-Identical Duplicate Consumer Photographs
*A Jaimes, S F Chang, *A Loui*
Columbia University, UNITED STATES
*Eastman Kodak Company, UNITED STATES
- 1A1.5 P0908** 21
Image Retrieval: A First Step for a Human Centered Approach
*A Guerin-Dugue, *S Ayache, *C Berrut*
Laboratory CLIPS, FRANCE
*Laboratoire CLIPS-IMAG, FRANCE
- 1A2 - Video Compression I**
- 1A2.1 P0442** 26
An Efficient Spatial Domain Error Concealment Method for H.264 Video
J H Zheng, L P Chau
Nanyang Technological University, SINGAPORE
- 1A2.2 P0403** 31
Block Image Retrieval Based on a Compressed Linear Quadtree
*Y K Chan, *C C Chang*
National Chung Hsing University, TAIWAN
*National Chung Cheng University, TAIWAN
- 1A2.3 P0259** 36
BAP Generation for Head Movement Based on MPEG-4 SNHC
S U Yoon, S Y Kim, Y S Ho
Kwangju Institute of Science and Technology, KOREA
- 1A2.4 P0300** 41
A Low Complexity 2-Power Transform for Video Compression
*S Lu, *K M Yu, *J Li, *S P Li*
University of Science and Technology of China, CHINA
*Microsoft Research Asia, CHINA

1A2.5 P0265	46
A Motion Compensation Method Using Least Squares Motion Estimation Filter in Wavelet Domain	
<i>M Y Tjoa, Y Morikawa, N Yamane, Y Imanishi</i>	
Okayama University, JAPAN	
1A2.6 P0330	51
A Platform Based MPEG-4 AVC Video Decoder with Block Level Pipelining	
<i>S H Wang, W H Peng, *Y W He, G Y Lin, C Y Lin, S C Chang, C N Wang, T H Chiang</i>	
National Chiao Tung University, TAIWAN	
*Tsinghua University, CHINA	
1A2.7 P0225	56
Minimization of Quantization Noise Amplification in 2-Channel IIR Bi-orthogonal Filter Bank	
<i>S Tripathi, *A Makur</i>	
Indian Institute of Science, INDIA	
*Nanyang Technological University, SINGAPORE	
1A2.8 P0219	61
Stereoscopic Video Coding Based on Global Displacement Compensated Prediction	
<i>H Z Jia, W Gao, *Y Lu</i>	
Chinese Academy of Sciences, CHINA	
*Harbin Institute of Technology, CHINA	
 1A3 - Image / Video Processing I	
1A3.1 P0210	66
Structure Tensor-Based Motion Field Classification and Optical Flow Estimation	
<i>H Y Wang, K K Ma</i>	
Nanyang Technological University, SINGAPORE	
1A3.2 P0660	71
Unified Feature Analysis in Different Compressed Domains	
<i>K M Au, N F Law, W C Siu</i>	
The Hong Kong Polytechnic University, HONG KONG S.A.R.	
1A3.3 P0732	76
Adaptive Color Filter Array Demosaicking Using Directional Spatial and Spectral Correlations	
<i>W M Lu, Y P Tan</i>	
Nanyang Technological University, SINGAPORE	
1A3.4 P0644	81
A Histogram Based Adaptive Vector Filter for Color Image Restoration	
<i>Z H Ma, H R Wu</i>	
Monash University, AUSTRALIA	
1A3.5 P0134	86
Markov Random Field Modeled Range Image Segmentation	
<i>X Wang, H Wang</i>	
Nanyang Technological University, SINGAPORE	
1A3.6 P0637	90
Novel H.26X Optimal Rate Control for Low-Delay Communications	
<i>C W Wong, O Au, B Meng, H K Lam</i>	
Hong Kong University of Science and Technology, HONG KONG S.A.R.	

1A3.7 P0775	95
Color Appearance-Based Approach to Robust Tracking and Recognition of Multiple People	
<i>J Tao, Y P Tan</i>	
Nanyang Technological University, SINGAPORE	
1A3.8 P0587	100
An Arbitrary Shaped Pattern Selection Algorithm for Very Low Bit-Rate Video Coding Focusing on Moving Regions	
<i>M Paul, M Murshed, L Dooley</i>	
Monash University, AUSTRALIA	
 1A4 - Acoustic Signal Processing	
1A4.1 P0882	105
Adaptive Beamformer with Recursively Updated Quadratic Constraints	
<i>H Q Dam, S Y Low, S Nordholm, *N Grbic</i>	
Western Australian Telecommunications Research Institute, AUSTRALIA	
*Blekinge Institute of Technology, SWEDEN	
1A4.2 P0881	110
Frame Erasure Pattern Feedback for Real-Time Perceptual Quality Estimation	
<i>B Rohani, *B Rohani, H J Zepernick</i>	
Western Australian Telecommunications Research Institute, AUSTRALIA	
*Curtin University of Technology, MALAYSIA	
1A4.3 P0915	114
Cancellation of Humming GSM Mobile Telephone Noise	
<i>I Claesson, *A Nilsson</i>	
Blekinge Institute of Technology, SWEDEN	
*Ericsson Mobile Platform, SWEDEN	
1A4.4 P0880	119
A Low Complexity Statistical Voice Activity Detector with Performance Comparisons to ITU-T / ETSI Voice Activity Detectors	
<i>A Davis, S Nordholm</i>	
Western Australian Telecommunications Research Institute, AUSTRALIA	
1A4.5 P0884	124
An Echo Path Variation Detector Based on Coherence	
<i>J Q Huo, S Nordholm, Z Q Zang</i>	
Western Australian Telecommunications Research Institute, AUSTRALIA	
1A4.6 P0883	129
Low Distortion SNR-Based Speech Enhancement Employing Critical Band Filter Banks	
<i>N Westerlund, J M De Haan, M Dahl, I Claesson</i>	
Blekinge Institute of Technology, SWEDEN	
 1A6 - Personal Communication Technologies	
1A6.1 P0871	134
Automated Cell Planning Based on Propagation Loss	
<i>X Y Wang, T Long, Y H Lee</i>	
Nanyang Technological University, SINGAPORE	
1A6.2 P0872	139
Reinforcement-Learning-Based Call Admission Control and Bandwidth Adaptation in Mobile Multimedia Networks	
<i>F Yu, V W S Wong, V C M Leung</i>	
University of British Columbia, CANADA	

1A6.3 P0874	144
Performance Study of GPRS/EDGE Systems for Voice and Data Transmission in Cellular Systems	
<i>C W Chen, M Yang, P H J Chong</i>	
Nanyang Technological University, SINGAPORE	
1A6.4 P0862	149
SIR-Dependent Scheduling and Static Power Preallocation for TDMA-Based Multi-Cell Wireless Packet Networks	
<i>R Veronesi, V Tralli</i>	
University of Ferrara, ITALY	
1A6.5 P0807	155
Design & Implementation of User Interface for Mobile Devices	
<i>D Goyal, P H J Chong, P Shum, Y C Tong, X Y Wang, Y X Zuo, H W Kuek</i>	
Nanyang Technological University, SINGAPORE	
1A6.6 P0873	160
Robust Adaptive Cluster Based Routing	
<i>Y S Tan, *W C Wong</i>	
National University of Singapore, SINGAPORE	
*Institute for Infocomm Research, SINGAPORE	
 1A7 - Optical Fiber Based Devices for Next Generation of Telecommunication Systems	
1A7.1 P0879	164
Investigations of Photonic Crystal Fiber Splicing	
<i>J H Chong, M K Rao, Y N Zhu, P Shum</i>	
Nanyang Technological University, SINGAPORE	
1A7.2 P0876	167
Fabrication of Optical Fiber Gratings and Device Applications for Communication and Sensing	
<i>Y Chung</i>	
Kwangju Institute of Science and Technology, KOREA	
1A7.3 P0878	171
Unique Dispersion Properties of Photonic Crystal Fibers	
<i>K Saitoh, M Koshiba</i>	
Hokkaido University, JAPAN	
1A7.4 P0877	176
Post-Processing of Photonic Crystal Fibres Using A CO₂ Laser Beam: A Step Towards Miniature Compact Fibre Devices	
<i>G Kakarantzas, *T A Birks, *P S J Russell</i>	
National Technical University of Athens, GREECE	
*University of Bath, UNITED KINGDOM	
1A7.5 P0875	178
All-Fibre Bragg Grating Filters and Lasers for Future Optical Networks	
<i>M Ibsen, J Nilsson, L B Fu, Y Jeong, J K Sahu, M R Mokhtar, C Alegria, D N Payne</i>	
University of Southampton, UNITED KINGDOM	

1B1 - Multimedia Systems & Services on the Web / Multimedia Services in Middleboxes

1B1.1 P0903 180
Towards Efficient Image Retrieval based on Multiple Features
B C Ooi, H T Shen
 National University of Singapore, SINGAPORE

1B1.2 P0902 186
Web Image Mining: Can We Gather Visual Knowledge for Image Recognition from The Web ?
K Yanai
 The University of Electro-Communications, JAPAN

1B1.3 P0901 191
Multimedia Web Services for Content Filtering, Searching and Digital Rights Management
Y L Wu, W C Lai, C W Chang, K T Cheng, E Chang
 University of California - Santa Barbara, UNITED STATES

1B1.4 P0900 197
Integrated Framework for the Visualization of Relational Databases and Related Web Content
T Sugibuchi, Y Tanaka
 Hokkaido University, JAPAN

1B1.5 P0709 202
A Programming Framework for Service Association in Ubiquitous Computing Environments
E Chen, Y C Shi, D G Zhang, G Y Xu
 Tsinghua University, CHINA

1B2 - Video Compression II

1B2.1 P0766 208
Transform Domain Overlapped Block Disparity Compensation in Wavelet Coding of Stereo Image Pairs
M Y Nayan, E A Edirisinghe, H E Bez
 Loughborough University, UNITED KINGDOM

1B2.2 P0580 213
Near Lossless Image Compression Using Lossless Hartley Like Transform
*P K Meher, T Srikanthan, *J Gupta, *H K Agarwal*
 Nanyang Technological University, SINGAPORE
 *Indian Institute of Technology, INDIA

1B2.3 P0537 218
Efficient Fast ME Predictions and Early-termination Strategy Based on H.264 Statistical Characters
J F Xu, Z B Chen, Y He
 Tsinghua University, CHINA

1B2.4 P0498 223
Adaptation of Visually Lossless Colour Coding to JPEG2000 in the Composite Colour Space
D M Tan, H R Wu
 Monash University, AUSTRALIA

1B2.5 P0564	230
Content Adaptive Frame Skipping for Low Bit Rate Video Coding	
<i>F Pan, *Z P Lin, X Lin, S Rahardja, *W Juwono, *F Slamer</i>	
Institute for Infocomm Research, SINGAPORE	
*Nanyang Technological University, SINGAPORE	
1B2.6 P0412	235
Noise Suppression for Shape-Gain Vector Quantization by Index Assignment Using Ant Colony Systems	
<i>C S Shieh, J S Pan, C J Su, B Y Liao</i>	
Kaohsiung University of Applied Sciences, TAIWAN	
1B2.7 P0460	239
Multi-Dimensional Integer Discrete W Transform	
<i>Y H Zeng, *G A Bi, *Z P Lin</i>	
The University of Hong Kong, HONG KONG S.A.R.	
*Nanyang Technological University, SINGAPORE	
1B2.8 P0229	244
A Circular Formulation of the Prime Length DHT Kernel and its FPGA Implementation	
<i>T Srikanthan, N Kumar, A Jhavar</i>	
Nanyang Technological University, SINGAPORE	
1B2.9 P0584	248
VLSI Design of a DWT/Modified Efficient SPIHT Based Image Codec	
<i>W B Huang, Y J Chang, A W Y Su</i>	
National Cheng Kung University, TAIWAN	
 1B3 - Image / Video Processing II	
1B3.1 P0668	253
A Temporal Error Concealment Algorithm for H.264 Based on Plane Estimation	
<i>J H Zheng, L P Chau</i>	
Nanyang Technological University, SINGAPORE	
1B3.2 P0754	258
Unsupervised Segmentation of Moving Object by Region-based MRF Model and Occlusion Detection	
<i>W Zeng, *W Gao</i>	
Harbin Institute of Technology, CHINA	
*Chinese Academy of Sciences, CHINA	
1B3.3 P0501	263
A Background Subtraction for a Vision-based User Interface	
<i>D Hong, W Woo</i>	
Kwangju Institute of Science and Technology, KOREA	
1B3.4 P0681	268
The Hybrid Video Error Concealment Algorithm with Low Complexity Approach	
<i>T H Tsai, Y X Lee</i>	
National Central University, TAIWAN	
1B3.5 P0659	272
Edge-Adaptive Color Reconstruction for Single-Sensor Digital Cameras	
<i>X M Wang, *W S Lin, P Xue</i>	
Nanyang Technological University, SINGAPORE	
*Institute for Infocomm Research, SINGAPORE	

1B3.6 P0624	277
Shape Representation of Clipart Image with Concave and Convex	
<i>C G Choi, Y S Chang, *S R Ryu, S H Kim</i>	
Kyungpook National University, KOREA	
*Chungwoon University, KOREA	
1B3.7 P0581	282
Fast Global Motion Estimation Using Iterative Least-Square Estimation Technique	
<i>G Sorwar, *M Murshed, *L Dooley</i>	
Southern Cross University, AUSTRALIA	
*Monash University, AUSTRALIA	
1B3.8 P0685	287
A New DCT-Based Just-Noticeable Distortion Estimator for Images	
<i>X H Zhang, *W S Lin, P Xue</i>	
Nanyang Technological University, SINGAPORE	
*Institute for Infocomm Research, SINGAPORE	
 1B4 - High Performance & Distributed Computing	
1B4.1 P0570	292
Interpolation of Poles in Asymptotic Waveform Evaluation	
<i>Y Cao, E C Tan</i>	
Nanyang Technological University, SINGAPORE	
1B4.2 P0297	296
Fault Tolerance Through Redundant COTS Components for Satellite Processing Applications	
<i>I V Mcloughlin, *V Gupta, *G S Sandhu, *S Lim, *T R Bretschneider</i>	
Tait Electronics Ltd, NEW ZEALAND	
*Nanyang Technological University, SINGAPORE	
1B4.3 P0734	300
Parallel Computaion for Multiple Sequence Alignments	
<i>Z H Du, F Lin</i>	
Nanyang Technological University, SINGAPORE	
1B4.4 P0512	304
Fault Tolerant Analysis of Secure Multicast Models	
<i>E Blessing R, R Uthariaraj V</i>	
Anna University, INDIA	
1B4.5 P0621	309
Minimizing Embedded Software Power Consumption Through Reduction of Data Memory Access	
<i>S Li, E Lai, *J Absar</i>	
Nanyang Technological University, SINGAPORE	
*ST Microelectronics Asia Pacific Pte Ltd, SINGAPORE	
1B4.6 P0516	314
Transaction Consistency in Replicated SIP Call Control Systems	
<i>M Bozinovski, T Renier, H Schwefel, R Prasad</i>	
Aalborg University, DENMARK	
1B4.7 P0126	319
An Investigation of Network Support for Grid Middleware and Applications	
<i>Y F Wong, *C Y Wong, *L H Nghoh, *W C Wong</i>	
National University of Singapore, SINGAPORE	
*Institute for Infocomm Research, SINGAPORE	

1B6 - Cellular and Mobile Networks

- 1B6.1 P0641** 324
Capacity Enhancement of CDMA Cellular Systems by Two-hop Relaying
K Yamamoto, S Yoshida
Kyoto University, JAPAN
- 1B6.2 P0674** 329
Constrained Power Control Scheme to Increase Capacity of Multi-service Systems
*C C Chai, Y H Chew, *Y Zhao, *K Li*
Institute for Infocomm Research, SINGAPORE
*National University of Singapore, SINGAPORE
- 1B6.3 P0481** 334
Analytical Modeling of Multiclass On/Off Services in Cellular CDMA Systems by Leaky Buckets
*T C Wong, *J W Mark, **K C Chua*
Institute for Infocomm Research, SINGAPORE
*University of Waterloo, CANADA
**National University of Singapore, SINGAPORE
- 1B6.4 P0636** 339
Seamless Roaming Service in Hybrid Network
S Park, D H Lee
SK Telecom, KOREA
- 1B6.5 P0233** 344
Performance Analysis of Variable Bit Rate Multiclass Services in a Multirate DS-CDMA System
*T C Wong, *J W Mark, **K C Chua, B Kannan*
Institute for Infocomm Research, SINGAPORE
*University of Waterloo, CANADA
**National University of Singapore, SINGAPORE
- 1B6.6 P0486** 349
Practical Design Considerations for Next Generation High-Speed Data Wireless Systems
C Y Huang, H Y Su
National Chiao Tung University, TAIWAN
- 1B6.7 P0232** 354
Resource Allocation for Video Services in Multirate DS-CDMA Cellular Networks with QoS Constraints
*T C Wong, *J W Mark, **K C Chua*
Institute for Infocomm Research, SINGAPORE
*University of Waterloo, CANADA
**National University of Singapore, SINGAPORE
- 1B6.8 P0808** 362
Embedded Human Interface Device for Voice and Data Communication
C L Chua, K S Koh, P H J Chong, P Shum, Y C Tong, X Y Wang, Y X Zuo, H W Kuek
Nanyang Technological University, SINGAPORE

1B7 - Radio Propagation / Satellite Systems

- 1B7.1 P0864** 369
On the Analysis of Single and Multiple Carrier WCDMA Systems with Polynomial Nonlinearities
*A Conti, D Dardari, *V Tralli*
University of Bologna, ITALY
*University of Ferrara, ITALY

1B7.2 P0845	376
An Improved UTD Propagation Model for Urban Cellular Communication System	
<i>L Jiang, S Y Tan</i>	
Nanyang Technological University, SINGAPORE	
1B7.3 P0513	381
The Accuracy of Geometric Channel Modeling Methods	
<i>Y F Chen, V K Dubey</i>	
Nanyang Technological University, SINGAPORE	
1B7.4 P0362	386
Wideband Tapped Delay Line Channel Model at 3.5GHz for Broadband Fixed Wireless Access System as Function of Subscriber Antenna Height in Suburban Environment	
<i>C L Hong, I J Wassell, *G E Athanasiadou, **S Greaves, **M Sellars</i>	
University of Cambridge, UNITED KINGDOM	
*University of Peloponnese, GREECE	
**Cambridge Broadband Ltd, UNITED KINGDOM	
1B7.5 P0853	391
The Design of Frequency Selective Surfaces (FSS) Using Real-coded Genetic Algorithm (RGA)	
<i>X F Luo, *A Y Qing, C K Lee</i>	
Nanyang Technological University, SINGAPORE	
*National University of Singapore, SINGAPORE	
1B7.6 P0865	396
Performance of a 2D-RAKE Receiver over Spatially Correlated, Slow Rayleigh Fading Channel	
<i>A W Umrani, V K Dubey</i>	
Nanyang Technological University, SINGAPORE	
1B7.7 P0686	401
Frequency Tracking and Acquisition with a Four-Quadrant arctan Phase Detector Based Digital Phase Locked Loop	
<i>S Kandeepan, S Reisenfeld</i>	
University of Technology Sydney, AUSTRALIA	
1B7.8 P0252	406
DTV Reception: Studies on Height Gain Measurements	
<i>J T Ong, H Yan, S V B Rao, S Ganesh</i>	
Nanyang Technological University, SINGAPORE	
1B7.9 P0427	411
Geometric Channel Model for Multipath Propagation in City-street Grids	
<i>Y F Chen, V K Dubey</i>	
Nanyang Technological University, SINGAPORE	
1C1 - Multimedia Server Technology	
1C1.1 P0891	416
Server Architectures for Interactive Multimedia Applications	
<i>F Li, *S S M Chan, *Q Li, *R W H Lau</i>	
Hong Kong Polytechnic University, HONG KONG S.A.R.	
*City University of Hong Kong, HONG KONG S.A.R.	
1C1.2 P0893	423
Managing Group Operations in IP Multicast	
<i>K Y Koh, H K Pung, Robin</i>	
National University of Singapore, SINGAPORE	

1C1.3 P0894	428
Adapting Images on Proxies for Small Form Factor Devices	
<i>X Xie, *X Fan, W Y Ma, *H Q Zhou</i>	
Microsoft Research Asia, CHINA	
*University of Science and Technology of China, CHINA	
1C1.4 P0889	433
H2O Clouds: Issues, Challenges and Solutions	
<i>S Ghandeharizadeh</i>	
University of Southern California, UNITED STATES	
1C1.5 P0888	438
Link Bandwidth Detection for Multimedia Streaming in a Distributed Server Environment	
<i>Y Liu, W J Jia, W Y Liu, C J Jiang</i>	
City University of Hong Kong, HONG KONG S.A.R.	
1C1.6 P0886	443
Scalable Multimedia Authentication	
<i>H Yu</i>	
Panasonic Information and Networking Technologies Laboratory, UNITED STATES	
1C1.7 P0885	448
Key Issues in Video Summarization and its Application	
<i>Y T Zhuang, F Wu, R G Xiao</i>	
Zhejiang University, CHINA	
1C1.8 P0890	453
Towards Better Performance Measurement of Web Servers	
<i>Y B Ling, *S G Chen, **X L Lin</i>	
Telcordia Technologies, UNITED STATES	
*University of Florida, UNITED STATES	
**City University of Hong Kong, HONG KONG S.A.R.	
 1C2 - Video Compression III	
1C2.1 P0852	458
Visually Lossless Adaptive Compression of Medical Images	
<i>D Wu, D M Tan, H R Wu</i>	
Monash University, AUSTRALIA	
1C2.2 P0185	464
Region-Based Guaranteed Image Quality in JPEG2000	
<i>T Trenchel, T R Bretschneider, C G Leedham</i>	
Nanyang Technological University, SINGAPORE	
1C2.3 P0770	469
Efficient Multithreading Implementation of H.264 Encoder on Intel Hyper-Threading Architectures	
<i>S Ge, X M Tian, Y K Chen</i>	
Intel Corp, CHINA	
1C2.4 P0123	474
On the Realtime Satellite Image Compression of X-Sat	
<i>R M Susilo, T R Bretschneider</i>	
Nanyang Technological University, SINGAPORE	

1C2.5 P0626	479
Partitioning Morphological Wavelet Image Coding	
<i>E Q Li, *H J Cui</i>	
Intel Corp, CHINA	
*Tsinghua University, CHINA	
1C2.6 P0823	483
Fast Video Coding at Low Bit-Rates for Mobile Devices	
<i>M Jindal, R S V Prasad, R Korada</i>	
Emuzed India Pte Ltd, INDIA	
1C2.7 P0698	488
Quadrature-Based Structure with Refined Context-Based Entropy Coding for Subband Image Compression	
<i>L C De Silva, *Z Y Li</i>	
Massey University, NEW ZEALAND	
*National University of Singapore, SINGAPORE	
1C2.8 P0143	493
A Novel Multi-View Video Coding Scheme Based on H.264	
<i>G P Li, Y He</i>	
Tsinghua University, CHINA	
1C2.9 P0112	498
Low Bitrate MPEG1 Layer III Encoder	
<i>K P Padhi, S Kumar, S George</i>	
ST Microelectronics Pte. Ltd., SINGAPORE	
1C3 - Image / Video Segmentation I	
1C3.1 P0274	503
Identification of Blur Support Size in Blind Image Deconvolution	
<i>L Chen, K H Yap</i>	
Nanyang Technological University, SINGAPORE	
1C3.2 P0284	508
A Hybrid Filter for the Cancellation of Mixed Gaussian Noise and Impulse Noise	
<i>R Li, Y J Zhang</i>	
Tsinghua University, CHINA	
1C3.3 P0159	513
An Improved Unequal Error Protection Technique for the Wireless Transmission of MPEG-4 Video	
<i>B Yan, K W Ng</i>	
The Chinese University of Hong Kong, HONG KONG S.A.R.	
1C3.4 P0144	518
Automatic Date Imprint Extraction from Natural Images	
<i>X Zhu, X G Lin</i>	
Tsinghua University, CHINA	
1C3.5 P0146	523
An Adaptive Spatial Filter for Additive Gaussian and Impulse Noise Reduction in Video Signals	
<i>Y Huang, L Hui</i>	
ST Microelectronics Asia Pacific Pte Ltd, SINGAPORE	
1C3.6 P0240	527
Color Image Segmentation with Watershed on Color Histogram and Markov Random Fields	
<i>S Y Dai, Y J Zhang</i>	
Tsinghua University, CHINA	

1C3.7 P0205	532
Design and Recognition of a Ring Code for AGV Localization	
<i>H Wang, N Liu</i>	
Nanyang Technological University, SINGAPORE	
1C3.8 P0199	537
Fast Image Restoration in Large Digital Photos	
<i>H Wu, *M J Li, *H J Zhang, Z K Liu</i>	
University of Science and Technology of China, CHINA	
*Microsoft Research Asia, CHINA	
1C3.9 P0167	541
Unsupervised Color Image Segmentation Based on Gaussian Mixture Model	
<i>Y M Wu, X Y Yang, K L Chan</i>	
Nanyang Technological University, SINGAPORE	
1C3.10 P0278	545
MPEG Transcoding for DVD Recording	
<i>L J Yuan, *H F Sun, W Gao</i>	
Chinese Academy of Sciences, CHINA	
*Mitsubishi Electric Research Laboratories, UNITED STATES	
 1C4 - Signal Processing for Communications	
1C4.1 P0206	549
Feed-Forward Carrier Frequency and Timing Synchronization for MSK Modulation	
<i>Y Yao, T S Ng</i>	
The University of Hong Kong, HONG KONG S.A.R.	
1C4.2 P0702	554
Design and Implementation of Timing Acquisition in IEEE 802.11a Wireless LANs	
<i>Z J Wang, B Zhao, F Jeung</i>	
Institute of Microelectronics, SINGAPORE	
1C4.3 P0437	559
Code Transformation of DF-Expression between Bintree and Quadtree	
<i>C C Chang, C F Li, *Y C Hu</i>	
National Chung Cheng University, TAIWAN	
*Providence University, TAIWAN	
1C4.4 P0803	564
Distortion and Designing of Signals Propagating in a Coaxial Cable	
<i>S K Foong, C H Lim</i>	
Natural Sciences NIE/NTU, SINGAPORE	
1C4.5 P0690	568
DSP Based Symbol Timing Estimation Techniques	
<i>S Kandeepan, S Reisenfeld</i>	
University of Technology Sydney, AUSTRALIA	
1C4.6 P0762	573
Design of CFAR Detector Combining with Frequency Estimation	
<i>Q Wang, Y X Yang, C R Wan</i>	
Nanyang Technological University, SINGAPORE	
1C4.7 P0388	578
A New Iterative Phase Tracking Scheme	
<i>C Vanstraceele, *J M Brossier, J P Barbot, **B Geller</i>	
SATIE (ENS Cachan/CNRS), FRANCE	
*LIS (INPG/CNRS), FRANCE	
**LETIC (Univ Paris12), FRANCE	

1C4.8 P0559	582
An Efficient Fine Granularity Scalable Coding Scheme Using Adaptive Leaky Prediction	
<i>Y L Gao, L P Chau</i>	
Nanyang Technological University, SINGAPORE	
1C4.9 P0736	587
Enhancing Interference Mitigation in Communication	
<i>W Y Leong, J Homer</i>	
The University of Queensland, AUSTRALIA	
1C4.10 P0595	592
Bounds on Frequency Estimation under Additive and Multiplicative Noise	
<i>Z Wang, S S Abeysekera</i>	
Nanyang Technological University, SINGAPORE	
1C6 - Equalization Techniques	
1C6.1 P0234	597
A Fast Algorithm for Single Block Finite Alphabet Based Channel Identification	
<i>D H Pham, J H Manton</i>	
The University of Melbourne, AUSTRALIA	
1C6.2 P0572	602
Decision Feedback Schemes for Enhanced Channel Estimation for Turbo Decoding in Rayleigh Fading Channels	
<i>W P Ang, H K Garg</i>	
National University of Singapore, SINGAPORE	
1C6.3 P0238	607
A First Order Statistical Method for Guard Interval Based Channel Identification and Source Recovery	
<i>D H Pham, J H Manton</i>	
The University of Melbourne, AUSTRALIA	
1C6.4 P0704	612
Equalization and Multi-user Detection for Stationary and Time varying Channels Using MRAN	
<i>R M Niranjan, V Sampath, S M S, *G Chandra M</i>	
Sri Jayachamarajendra College of Engineering, INDIA	
*National Aerospace Laboratories, INDIA	
1C6.5 P0712	617
Time Reversal Space Time Block Coding with Channel Estimation Errors	
<i>K Mehrotra, I V Mccloughlin</i>	
Tait Electronics Ltd, NEW ZEALAND	
1C6.6 P0560	621
An Iterative Detection for MIMO Wireless Systems Using Projection and Hard-Decision with a Reliability Measure	
<i>J Choi</i>	
The University of New South Wales, AUSTRALIA	
1C6.7 P0244	626
A Novel RMS Delay-Spread Estimation Technique for Wireless OFDM Systems	
<i>C Athaudage, *D Jayalath</i>	
The University of Melbourne, AUSTRALIA	
*Australian National University, AUSTRALIA	

1C6.8 P0869	631
A Novel Adaptive Equalization and Cancellation Algorithm for Direct Conversion Receivers	
<i>Y J Zheng, M Z Cao, *H K Garg</i>	
Institute of Microelectronics, SINGAPORE	
*National University of Singapore, SINGAPORE	
 1C7 - Spread Spectrum Techniques	
1C7.1 P0527	636
Optimal Uplink Sequence Assignment for Multicarrier CDMA Systems in Rayleigh Fading Channels	
<i>T M Lok</i>	
The Chinese University of Hong Kong, HONG KONG S.A.R.	
1C7.2 P0207	641
A Novel Approach for Interference Cancellation of DS-CDMA Systems Using Projected Successive Overrelaxation Algorithm	
<i>K Yang, *A S Madhukumar, **F Chin</i>	
National University of Singapore, SINGAPORE	
*Nanyang Technological University, SINGAPORE	
**Institute for Infocomm Research, SINGAPORE	
1C7.3 P0218	645
On Generalized Orthogonal Spreading Sequences for Quasi Synchronous CDMA System	
<i>F X Zeng, *L J Ge</i>	
Chongqing Communication Institute, CHINA	
*Chongqing Sinotel Digital Communications & Signal Processing Institute, CHINA	
1C7.4 P0208	650
Cyclic Prefix Assisted Code Division Multiple Access Communications System: A New Approach for High-Rate Data Transmission	
<i>A S Madhukumar, *F Chin, *Y C Liang, **K Yang</i>	
Nanyang Technological University, SINGAPORE	
*Institute for Infocomm Research, SINGAPORE	
**National University of Singapore, SINGAPORE	
1C7.5 P0253	655
Estimating the Spatial Time Correlation of Signals from Two Obstructed Mobile Transceivers	
<i>H King</i>	
RMIT University, AUSTRALIA	
1C7.6 P0281	658
Performance Analysis for Downlink MC-CDMA Systems with Space-time Block Codes in Frequency-selective Rayleigh Fading Channels	
<i>Z H Hou, V K Dubey</i>	
Nanyang Technological University, SINGAPORE	
1C7.7 P0436	663
Channel Identifiability of OFDM-CDMA System Based on Subspace Method	
<i>Y H Zeng, T S Ng</i>	
The University of Hong Kong, HONG KONG S.A.R.	
1C7.8 P0307	668
Simplified BER Formulations for MC-CDMA Systems with Orthogonality Restoring Combining	
<i>K L Zhang, Y L Guan</i>	

Nanyang Technological University, SINGAPORE

1C7.9 P0435 673
Performance of Carrier Frequency Offset Estimation in MC-CDMA Systems over Frequency Selective Fading Channels
F Yang, K H Li, K C Teh
Nanyang Technological University, SINGAPORE

1C7.10 P0623 678
Performance Analyses of Coherent Fast Frequency-Hopping Spread-Spectrum Systems with Partial Band Noise Jamming and AWGN
J J Kang, K C Teh
Nanyang Technological University, SINGAPORE

Author Index A1

Volume II

2A1 - Multimedia Applications I

2A1.1 P0752 683
Recovery of Reflectance Properties for Merging Multiple 3D Scans
*X Y Jiang, C F Chang, W Y Ho, S T Chen, *I C Chang*
National Tsing Hua University, TAIWAN
*Industrial Technology Research Institute, TAIWAN

2A1.2 P0729 688
Computational Video Editing Model based on Optimization with Constraint-Satisfaction
R Ogata, Y Nakamura, Y Ohta
University of Tsukuba, JAPAN

2A1.3 P0858 694
Facial Expression Cloning: Using Expressive Ratio Image and FAP to Texture Conversion
*P W Hsu, *Y J Chang, C K Hsieh, Y C Chen*
National Tsing Hua University, TAIWAN
*Industrial Technology Research Institute, TAIWAN

2A1.4 P0170 699
An Effective Eye Gaze Correction Operation for Video Conference Using Anti-Rotation Formulas
B Yip, J S Jin
University of Sydney, AUSTRALIA

2A1.5 P0324 704
On the Implementation of Melody Recognition on 8-Bit and 16-Bit Microcontrollers
J S R Jang, Y S Jang
Tsing Hua University, TAIWAN

2A1.6 P0151 709
Towards a Conversational-Driven Scenario-Based Multimedia Training System
*B Tootell, R K Y Li, *D Graham, *P Baiguera*
Monash University, AUSTRALIA
*Victoria Police, AUSTRALIA

2A1.7 P0688 715
Rendering Spatial Sound on Headsets for Five-Channel Audio
*Y C Shen, *S C D You*
National Chiao-Tung University, TAIWAN

Ex. 1007A

A Platform-Based MPEG-4 Advanced Video Coding (AVC) Decoder with Block Level Pipelining

Shih-Hao Wang, Wen-Hsiao Peng, Yuwen He¹, Guan-Yi Lin, Chen-Yi Lin, Shih-Chien Chang, Chung-Neng Wang, and Tihao Chiang^{2*}

Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan

¹Department of Computer Science and Technology, Tsinghua University, Beijing 100084

Abstract

We present a baseline MPEG-4 AVC (Advanced Video Coding) decoder based on an optimized platform-based design methodology. With this methodology, we jointly optimize the software and hardware design of the decoder. Overall decoding throughput is increased by synchronizing the software and the dedicated co-processors. The synchronization is achieved at macroblock-level pipelining. In addition, we optimize the decoder software by enhancing the frame buffer management, boundary padding, and content aware inverse transform. To speed up motion compensation and inverse transform, which are the most computationally intensive modules, two dedicated acceleration modules are realized. For comparison, the proposed prototype decoder and MPEG-4 AVC reference decoder are evaluated on an ARM platform, which is one of most popular portable devices. Our experiments show that the throughput of the MPEG-4 reference decoder can be improved by 6 to 7 times. On an ARM966 board, the optimized software without hardware acceleration can achieve a decoding rate up to 5 frames per second (fps) for QCIF video sequences. With the dedicated accelerators, the overall throughput is increased by about 30% to reach 6.6 fps on the average and is up to 10.3 fps for slow motion video sequences.

1. Introduction

Multimedia services over mobile network are becoming important with the pervasive presence of the Internet and wireless devices. For this kind of application, MPEG-4 Advanced Video Coding [7] provides better image quality due to its significant improvement on the coding efficiency as compared to that of MPEG-4 visual simple profile. Since some of the AVC modules consume considerable power, it is challenging to realize real-time AVC encoder and decoder. Several AVC codec [8][9] are optimized for general purpose processors that have sufficient computation power, large memory, special instruction sets, and wide buses to facilitate the speedup of AVC modules. But the performance of the optimized AVC modules is seriously constrained by the low computation power and small memory for mobile devices to achieve the least service requirement of QCIF 7.5 fps. To address such constraints with limited resource, we adopt platform based design method to increase overall throughput.

The design methodology using existing platforms for a multimedia system on a single chip has been widely adopted as in [2]-[6]. The design in [6] is optimized for TI C62x DSP and the design in [5] is for ARM7 TDMI processor. There are other designs [2-4] that adopt a physical ASIC design methodology with maximum performance but longer design cycle. On the other hand, the former case of using existing platform has more flexibility on implementation and algorithm level verification but lacks high performance and cost effectiveness. Thus, there is a tradeoff between the efficiency and the design cycle. To address the problems of extended design cycle and less performance in the case of general purpose processor, we propose an optimized platform-based architecture with macroblock (MB) level pipelining to implement a baseline MPEG-4 AVC decoder. On this platform, a general-purpose processor controls the decoding processes and dedicated acceleration modules are used to speed up the computationally intensive tasks. The accelerators are realized with programmable logic modules to retain the flexibility. In addition, a specific system bus is used to remove the bottleneck for data communications between modules. Subsequently, the proposed architecture is evaluated on an ARM966 platform. Our results show that the optimized software without hardware acceleration can achieve a decoding rate up to 5 frames per second (fps) for the video sequences in QCIF resolution. With the new dedicated accelerators, the decoding throughput is increased by about 30% to reach 6.6 fps on the average and is up to 10.3 fps for slow motion video sequences.

2. MPEG-4 AVC Decoder

Fig. 1 shows an MPEG-4 AVC decoder architecture where the decoding tasks are partitioned into four main modules as follows.

1. Context-adaptive variable length decoding (CAVLD).
2. Inverse quantization and inverse 4x4 discrete cosine transform (IQ-IDCT).
3. Motion compensation (MC).
4. Reconstruction and loop filtering.

The decoding is performed by first parsing the compressed bit-stream by CAVLD. After the parsing, the quantized prediction residues and MB side information including the MB type (MB_Type), the prediction mode, and the motion vector difference (MVD) are extracted. The extracted MB_type controls the switches S_0 and S_1 in Fig. 1 to determine the prediction type. The intra prediction values are derived based on the neighboring pixels without loop

¹ Yuwen He : ywhe@mail.tsinghua.edu.cn

² Tihao Chiang : tchiang@mail.nctu.edu.tw

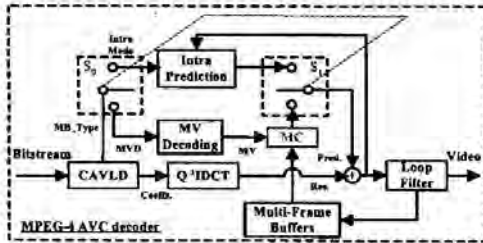


Fig. 1. ARM-based architecture of MPEG-4 AVC decoder

filtering for an intra MB and the inter prediction values are generated from motion compensated pixels for an inter MB. The addition of the prediction and the decoded residuals produces the reconstructed frame. After the reconstruction, a loop-filter is applied to remove blocking artifacts and the filtered results are put into the frame buffers for future reference.

3. ARM Platform Based Architecture

In this paper, we adopt a platform based design methodology to construct an optimized AVC decoder with a novel scheduling to achieve MB-level pipelining.

3.1 Platform-Based Design Methodology

The platform-based design methodology has been widely adopted to solve complicated system-level designs of a multimedia system on a single chip. The platform-based design could be defined in following two ways [1], [10]. First is reuse of architecture of hardware and software blocks. Second is the construction of a system with stable microprocessor, memory hierarchy, interconnecting bus, and peripherals. The construction has the abilities to cover rapid extension, feasible customization for a wide range of applications, and a short time-to-market. In addition, the platform-based design can improve the yield in circuit design. In short, the platform-based design [1] has the multiple advantages including efficient system-level design methodology, short time-to-market, reusability of software and hardware IP blocks, feasible chip integration, etc. Thus, the platform-based design methodology is adopted for this project.

Some systems [2]-[4] are developed on a circuit design level and some systems [5]-[6] are built on existing hardware platforms. In [2], a videophone system compliant with H.263 is developed where the system is partitioned into hardware modules and software modules based on complexity analysis. Several microcontrollers are used to manage data and functional flow. The dedicated hardware is used to improve computationally intensive parts. In addition, the modules work with MB-level pipelining. In [3] the MPEG-2 encoding processes is partitioned into 3 layers including processing control, video processing, and data buffering. A RISC processor controls and pipelines the 3-layer functional modules in MB-level. In [4], a multi-core SoC architecture is proposed for MPEG-4 streaming video. Based on the application profile characteristics, the task scheduling of the decoding processes is optimized by a MB engine. In addition, the SoC architecture has a global controller using a RISC processor and a computation

Table 1. MPEG-4 AVC baseline profile and level 1.

Max MB processing rate	1485
Max picture size	176x144
Max bit rate	64kbit/s
Max # of reference frames	5
Max horizontal and vertical MV range (full pels)	-16~16

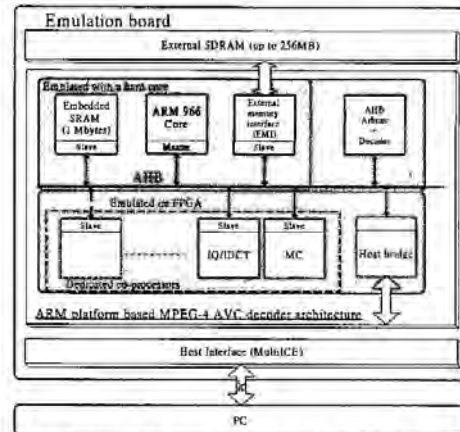


Fig. 2. ARM platform based MPEG-4 AVC decoder architecture and emulation environment.

accelerator with a DSP. On a multi-core platform, all processor cores are communicated via ARM Master Bus Architecture (AMBA). In summary, the platform-based design methodology has been widely adopted to construct the codec of H.263 and MPEG-2/4 standards as discussed [2]-[6]. Since the complexity of the MPEG-4 AVC codec is much higher than that of the aforementioned coding standards, it is challenging to build an AVC decoder on existing platforms.

3.2 Proposed Architecture

Fig. 2 shows our emulation configuration to test our new architecture for the MPEG-4 AVC decoder. The proposed architecture is compliant with the MPEG-4 AVC baseline profile of level 1. Table 1 summarizes its main parameters. The architecture is emulated on an ARM development board. Since the ARM platform is one of the most typical portable devices and takes almost 75% of the market for the mobile devices. In addition, the ARM platform provides a general purpose ARM966 CPU core for data flow control and a logic module for multiple dedicated accelerators. The ARM also provides an industry standardized AHB for high-speed computation and emulations.

The central part of Fig. 2 depicts our ARM platform based MPEG-4 AVC decoder architecture. Mainly, it includes an ARM966 CPU core, an embedded SRAM, multiple dedicated accelerators, an external memory interface as well as a 32-bit wide AHB bus connecting all functional blocks.

The ARM966 CPU acts as a master on the AHB bus and controls the synchronization among all functional blocks. All the remaining functional blocks that respond to the requests from CPU are slaves. Specifically, the dedicated accelerators are used to speed up computation or reduce

Table 2. Key operations for AVC decoding modules.

Modules	MC	IQ-IDCT	CAVLD	Loop Filter
Operation	Mul., Add, Shift, MemA	Add, Shift	Branch, MemA	Add, MemA

Mul.: multiplication; MemA: memory access

memory access. The firmware of the accelerators and the software for the decoding modules are stored in the embedded SRAM. In addition to the embedded memory, our decoder also requires external memory for frame buffering. The external memory is accessed via an external memory interface.

4. Design Methodology

4.1 Software and Hardware Partition

In the ARM platform based design, a task can be done with either the software executing on CPU or the dedicated hardware running in parallel with CPU. Thus, to optimize the overall performance through parallel processing, it's challenging to partition the tasks to separately match CPU capability and the dedicated accelerators.

Computational characteristic is a good criterion for task partition [2]-[4]. The modules with regular and computational intensive tasks are perfect for hardware implementation and the modules with lots of branches are more suitable for software realization. Table 2 illustrates the kernel operations of each AVC decoding module. Most modules except the IQ-IDCT require a great amount of memory access (MemA). In addition, the MC and loop filter require intensive arithmetic operations for interpolation and filtering, respectively. The CAVLD uses lots of branching instructions for context-adaptive table switching. Thus, to optimize the performance, the MC, loop filter, and IQ-IDCT are implemented in hardware and the CAVLD is realized in software.

Fig. 3 shows the decoding profiling in relative execution time for different modules. From the Amdahl's law and the observations of Fig. 3, we reduce the computational loads by adding two dedicated accelerators for the MC and IQ-IDCT, respectively. The remaining coding modules are implemented and optimized in software.

4.2 Software and Hardware Synchronization

To save the buffers for intermediate data and maximize the throughput, module synchronization is required. In Fig. 4, we design a scheduling approach for MB-level pipelining based on data dependency and working load distribution. The subscript n denotes the MB index. Basically, a MB is decoded through the three stages including (1) CAVLD; (2) IQ-IDCT and MC; (3) reconstruction and loop filtering. Without a dedicated accelerator for the loop filtering, an ideal pipelining schedule in Fig.4 is changed into that in Fig. 5, where the CAVLD, reconstruction, and loop filtering share one CPU.

In Fig.5, the data dependency at the same stage may decrease the overall throughput. To address this problem,

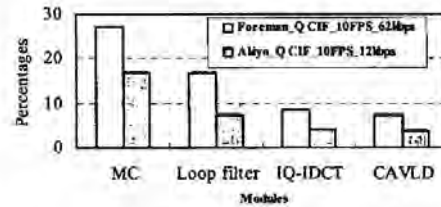


Fig. 3. Decoding profiling for AVC on ARM 966 CPU.

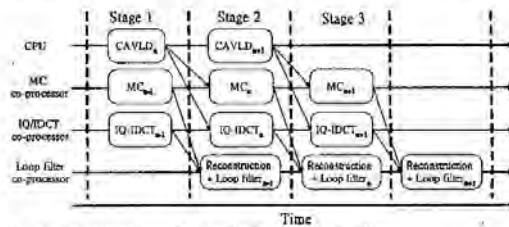


Fig. 4. Illustration of ideal MB level pipeline.

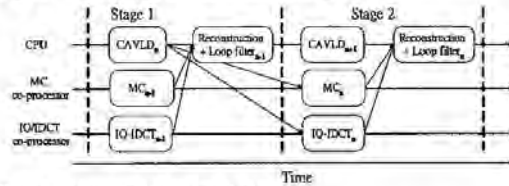


Fig. 5. Modified MB level pipeline.

we maximize the CPU cache performance to have the throughput close to that of the ideal scheduling in Fig.4. Thus, the modified pipeline schedule performs the reconstruction and loop filtering of a MB once CPU has received the results of IQ-IDCT and MC from the dedicated accelerators.

Fig. 6 shows the flow chart to synchronize CPU with the two accelerators at MB-level. At the beginning of each stage,

1. Send the data to the MC accelerators for motion compensation of the n -th MB.
2. Send the data to the IQ-IDCT accelerators for IQ-IDCT of the n -th MB.
3. Proceed to decode the $(n+1)$ -th MB header on the CPU.
4. Check the ready signals from the two accelerators.
5. If the data in the two accelerators are both ready, pass the data through the reconstruction and loop filtering to reconstruct the n -th MB. Otherwise, go to Stage 4.

4.3 Software Optimization

To retain the flexibility and portability across different platforms, our software optimization is mainly from algorithm level improvement. Our optimization techniques are as follows.

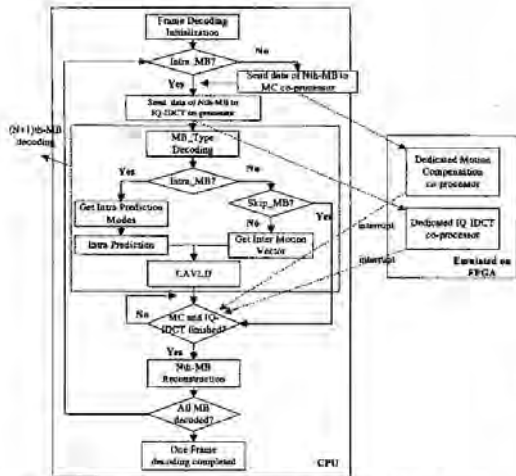


Fig. 6. Flow chart for the MB level pipeline in software.

A. Frame buffer management

We improve the frame buffer management to save execution time for memory copy as follows.

1. The frame copy is done by swapping the pointers of frame buffers.
2. The motion compensated frame or intra prediction pixels are stored directly in the decoded frame buffer to avoid memory copy.
3. The decoded residuals are stored in the frame buffers with 4-byte alignment to match the 32-bit access architecture for ARM.

B. Boundary padding for motion compensation

Costly branching instructions for boundary checking can be saved if we perform padding in advance. Since unconstrained motion vector and the 6-tap filter [1, -5, 20, 20, -5, 1] are used for MC with half-pel interpolation, the locations of the reference pixels could exceed the frame boundaries. To avoid costly branching instructions, the reference frame are padded before MC. The padding length for the left and upper boundaries is (16+2) points while the padding length for the right and bottom boundaries is (16+3) points. In addition to padding, we also speed up the interpolation by replacing the multiplication for fixed filter tap weight with a table look-up.

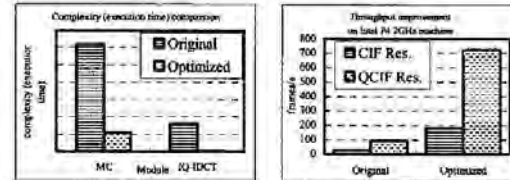
C. Content aware IQ-IDCT

Due to correlation removal from inter frame prediction; most DCT coefficients after quantization are zero. Table 3 shows the percentage of all zero blocks in fast motion and slow motion sequences. For fast motion sequences, about 25% of DCT blocks are classified as the all zero case. For slow motion sequences, the percentage is increased to 47%. Besides, we also find that a large percentage of DCT blocks only have one DC coefficient. Thus, we can classify DCT blocks into three cases and conduct IQ-IDCT differently as follows:

1. For the blocks with nonzero AC coefficients, perform the normal IQ-IDCT.

Table 3. Percentage of all zero blocks at 64 kbits/sec.

Sequence(QCIF)	Fast motion	Slow motion
	Foreman	Mother-Daughter
All zero blocks	24.8%	47.0%



(a) Complexity

(b) Decoding frame rate

Fig. 7. Comparison of MPEG-4 AVC decoders with and without optimization

2. For the blocks with only nonzero DC coefficient, perform only the inverse DC transform.
 3. For the blocks with all zeros, skip the transform.
- With our optimization, the entire code size is reduced to 80 kilo-bytes and the complexity for the MC, loop filter and IQ-IDCT are greatly reduced. Fig. 7 (a) shows the complexity comparison in execution time with and without our optimization. Further, Fig. 7 (b) shows the decoding throughput improvement (on an Intel P4 2GHz PC). With our optimization, the overall throughput is improved by 6 to 7 times on the average.

4.4 Dedicated Accelerator Design

By allocating dedicated co-processors with only regular and computation intensive tasks, we design the co-processors as simple as possible. With the MB level scheduling, each co-processor operates at the granularity of one MB.

A. MB-based MC

To speed up interpolation, we design a dedicated co-processor in Fig. 8. For a MB interpolation, the co-processor uses a local memory to store 1500 integer pixels and two interpolation engines for parallel processing of the luminance and chrominance components. Each engine consists of multiple multipliers and accumulators. Particularly, the multiplier is implemented in a hardwired manner to maximize performance. To get one interpolated MB, the intermediate results (output) of row and column filtering in each engine may be fed back to conduct the filtering of another columns and rows.

The AVC specification uses variable block size motion compensation. The minimum granularity for motion compensation is a 4x4 block. Therefore, our interpolation engine is designed for a 4x4 block. The interpolation of each MB takes 16 iterations. In the worst case, our design takes 1280 cycles to interpolate one MB. As operating at 10 MHz, the throughput for each MB is 7812 MB/sec, which is over the requirements in Table 1. The area used is 11172 gates.

B. MB based IQ-IDCT

The IQ-IDCT involves two main steps. The first step is to store the input data of a MB in local memory with the size 3 kbits. The second step is to perform the 4x4 integer IDCT,

which consists of two multiplication-free transforms and one transpose operation. In addition, the IQ task is done by a hardwired multiplier. In our implementation, we integrate three different types of inverse transform for luminance 4x4 DC block, chrominance 2x2 DC block, and luminance 4x4 block. In the worst case, the IQ-IDCT takes 210 cycles for a MB. When operating at 10 MHz, the throughput is 47619 MB/sec, which is over the requirement of the AVC decoder in Table 1. The area used is 11489 gates.

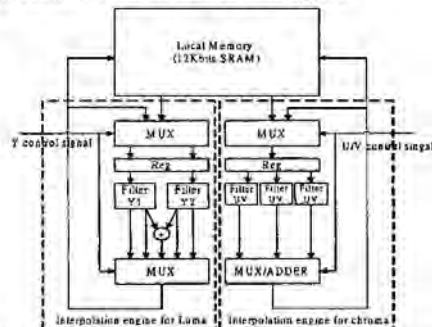


Fig. 8. Interpolation architecture for quarter-pel motion compensation

5. Experiment Results and Analysis

A. Decoding Throughput

Our decoding system is emulated with the configuration in Fig. 3. The ARM966 CPU (system clock running at 130MHz), embedded SRAM (1 Mbytes), and external memory interface are replaced with a dedicated hard core while the other parts including the dedicated accelerators are implemented on a FPGA module (running at 10MHz). In addition, the AHB bus is running at 33MHz.

Table 4 lists the decoding throughput in frames per second (fps) on ARM966 CPU. With our software only optimization, the decoding speed is about 2.3 to 8.1 fps. With hardware acceleration, the throughput is increased by 30%~60%. The overall throughput ranges from 3.3 to 10.3 fps. As shown in Table 4, the decoding throughput is sequence dependent. Our decoder performs better for slow motion sequences, which have more zero DCT blocks and high probability of using integer pixel resolution MC. The decoding rate is increased due to less computation for the MC and IDCT-IQ modules with slow motion sequences. The improvement ratio using accelerators for decoding slow motion sequences is smaller, which is consistent of the observations of Amdahl's law and the results in Table 4.

To analyze the performance of the proposed method, the ideal upper bound is reached as 7.7 fps¹ following Amdahl's law. (The dedicate hardware occupies almost 35% for Foreman sequence according to Fig. 3) The performance gap can be explained with the overhead for synchronization and bus communication. Due to the sub-optimal synchronization scheme, it causes some overhead and makes the decoding rate slower than the ideal 7.7 fps.

$$fps_{ideal} = fps_{sw} \times \frac{100\%}{100\% - P_{MC+IDCT}(\%)} = 7.7(fps)$$

Table 4. Performance of our prototype decoder.

QCIF	Bit-rate (kbits/s)	SW only (fps)	SW+HW (fps)	Throughput Speedup
Container	13.15	6.8	8.6	26%
Akiyo	8.67	8.1	10.3	27%
Coastguard	58.69	2.3	3.6	44%
Mother_Daughter	14.43	4.5	6.8	50%
Foreman	42.47	2.4	3.7	56%

Frame size = QCIF, Group of picture= I1 + 74P, Qp = I(30) P(31), frame rate = 7.5 fps, reference frame number = 5

6. Conclusions

In this paper, we have proposed a MB-level pipelining architecture for a MPEG-4 AVC baseline decoder based on both platform-based design methodology and application specific circuit design methodology. The platform design methodology shortens the design cycles and the application specific circuit design methodology increases the flexibility with accelerators. The application specific circuit design methodology includes the task partitioning and scheduling in the MB-level to enhance the overall decoding throughput. With the task partitioning, the software parts control the branching data flow and the hardware accelerators speed up the regular and computationally intensive modules. In addition, without branching operations, the accelerator design is simplified. With the platform-based design methodology, the software procedures and hardware modules retain a high degree of reusability. The proposed MB-level pipelining achieves significant improvement in speed using both software and hardware co-design.

Reference

- [1] G. Martin, "A design chain for embedded systems," *IEEE Computer Magazine*, March, 2002.
- [2] M. Harrand, et al., "A single chip CIF 30-Hz, H.261, H.263, and H.263+ video encoder/decoder with embedded display controller," *IEEE Journal of Solid State Circuits*, vol. 34, pp.1627-1633, Nov. 1998.
- [3] M. Ikeda, et al., "SuperEnc: MPEG-2 video encoder chip," *IEEE Micro magazine*, July 1999.
- [4] M. Berekovic, et al., "Multicore system-on-chip architecture for MPEG-4 streaming video," *IEEE Trans. On Circuit and System for Video Technology*, vol. 12, no. 8, August 2002.
- [5] K. Ramkishor and V. Gunashree, "Real time implementation of MPEG-4 video decoder on ARM7TDMI," *Proc. IEEE International Symposium on Intelligent Multimedia, Video, and Speech Processing*, May 2001.
- [6] Minhua Zhou and Raj Talluri, "DSP-based real time video decoding," *Proc. IEEE International Conference on Consumer Electronics*, pp. 296-297, 1999.
- [7] Joint Video Team (JVT) of ISO/IEC MPEG&ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification", JVT-G050, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Mar. 2003.
- [8] J. Zhang, Y. He, S. Yang and Y. Zhong, "Performance and Complexity Joint Optimization for H.264 Video Coding", in *Proc. of IEEE ISCAS*, May 2003.
- [9] V. Lappalainen, et al., "Performance of H.26L video encoder on general purpose processor," *Journal of VLSI Signal Processing*, pp. 239-249, 2003.
- [10] T. Givargis and F. Vahid, "Platune: A tuning framework for system-on-chip platforms," *IEEE Trans. on Computer-Aided-Design of Integrated Circuits and Systems*, Nov. 2002.

Ex. 1007B

IEEE Personal Account

CHANGE USERNAME/
PASSWORD

Purchase Details

PAYMENT OPTIONS
VIEW PURCHASED
DOCUMENTS

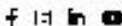
Profile Information

COMMUNICATIONS
PREFERENCES
PROFESSION AND
EDUCATION
TECHNICAL INTERESTS

Need Help?

US & CANADA: 1-800-
678-4333
WORLDWIDE: +1-732-981-
0000
CONTACT & SUPPORT

Follow



About IEEE Xplore | Contact Us | Help | Accessibility | Terms of Use | Nondiscrimination Policy | IEEE Ethics Reporting | Stemup
IEEE Privacy Policy
A public charity, IEEE is the world's largest technical professionals organization dedicated to advancing technology for the benefit of humanity.
©Copyright 2020 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies.

Ex. 1007C

← → ↻ <https://web.archive.org/web/20010207003457/http://ieeexplore.ieee.org/lpdocs/epic03/> 133% ☆

<http://ieeexplore.ieee.org/lpdocs/epic03/> Go NOV FEB APR
115 captures
7 Apr 2000 – 2 Dec 2024 2000 2001 2004 Abc

About IEEE | IEEE Memberships | Products and Services | Conferences | IEEE Organizations | News | Home

IEEE Xplore™

Home

Help FAQ Terms Technical Update

Welcome to IEEE Xplore™

- Home
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

On Sunday, 18 February, from 6 AM through 2 PM Eastern Standard Time, IEEE will be conducting planned maintenance which may cause temporary disruption of user access to the IEEE Xplore site. To minimize user inconvenience this maintenance has been scheduled for a period of minimal activity on the site.

IEEE Xplore provides full-text access to IEEE transactions, journals, magazines and conference proceedings published since 1988 and all current IEEE standards.

A Special Message to IEEE Members and Customers
Recently, users of the IEEE Xplore site have been experiencing intermittent access problems. Resolving these issues is IEEE's highest priority. For more information on the situation and steps we are taking [Please see our Technical Update message.](#)

More About...

- IEEE/IEE Electronic Library (IEL)
- IEEE All-Society Periodicals Package (ASPP)
- IEEE Conference Proceedings Order Plan (POPs)
- IEEE Spectrum Magazine
- The Proceedings of the IEEE
- IEEE Standards Online

The Wayback Machine - <https://web.archive.org/web/20010207003457/http://ieeexplore.ieee.org:80/lpdocs/epic03/>

[About IEEE](#) | [IEEE Memberships](#) | [Products and Services](#) | [Conferences](#) | [IEEE Organizations](#) | [News](#) | [Home](#)

[Home](#)[Help](#) [FAQ](#) [Terms](#) [Technical Update](#)**Welcome to IEEE Xplore™**

- [Home](#)
- [Log-out](#)

Tables of Contents

- [Journals & Magazines](#)
- [Conference Proceedings](#)
- [Standards](#)

Search

- [By Author](#)
- [Basic](#)
- [Advanced](#)

Member Services

- [Join IEEE](#)
- [Establish IEEE Web Account](#)

POWERED BY
LIVEPAGE

POWERED BY
eRights

On Sunday, 18 February, from 6 AM through 2 PM Eastern Standard Time, IEEE will be conducting planned maintenance which may cause temporary disruption of user access to the IEEE Xplore site. To minimize user inconvenience this maintenance has been scheduled for a period of minimal activity on the site.

IEEE Xplore provides full-text access to IEEE transactions, journals, magazines and conference proceedings published since 1988 and all current IEEE standards.

A Special Message to IEEE Members and Customers

Recently, users of the IEEE Xplore site have been experiencing intermittent access problems. Resolving these issues is IEEE's highest priority. For more information on the situation and steps we are taking [Please see our Technical Update message](#).

IMPORTANT - NO ROBOTS PLEASE!

This site does not permit the use of intelligent agents ("robots") or other means of systematically downloading content. Attempting to download content through these mechanisms places significant strain on the site and can harm all users. Robots must not be used under any circumstances. [More...](#)

FREE TO ALL: Access to tables of contents of IEEE transactions, journals, magazines, conference proceedings and standards.

IEEE MEMBERS: You can search and access all IEEE abstract/citation records and articles from IEEE Spectrum Magazine free with IEEE membership. You may also access your online personal subscriptions (known as OPeRA) here. You must have an active Web account to access these services. Log-in or go to "Establish IEEE Web Account" to set up your account, and then log in.

CORPORATE, GOVERNMENT AND UNIVERSITY SUBSCRIBERS: You can search and access abstract/citation records, and full-text documents of the IEEE online publications to which your institution subscribes.

More About...

- [IEEE/IEE Electronic Library \(IEL\)](#)
- [IEEE All-Society Periodicals Package \(ASPP\)](#)
- [IEEE Conference Proceedings Order Plan \(POPs\)](#)
- [IEEE Spectrum Magazine](#)
- [The Proceedings of the IEEE](#)
- [IEEE Standards Online](#)

**For Current
OPeRA and
Bibliographies
Online Users**

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#)
[Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Establish a Web Account](#)

Copyright © 2000 IEEE -- All rights reserved

Ex. 1007D

Copyright

United States Copyright Office

Try the [Copyright Public Records System \(CPRS\)](#) pilot with enhanced search features and filters.

[Help](#)[Search](#)[History](#)[Titles](#)[Start Over](#)

Public Catalog

Copyright Catalog (1978 to present) at DC4

Search Request: Left Anchored Title = ICICS-PCM 2003

Search Results: Displaying 1 of 1 entries

[◀ previous](#) [next ▶](#)[Labeled View](#)

ICICS-PCM 2003 : proceedings of the 2003 joint conference of the 4th...

Type of Work: Text

Registration Number / Date: TX0006050947 / 2004-10-15

Application Title: 2003 joint conference of the 4th Int'l conf. on communications & signal processing & 4th Pacific-Rim conference on multimedia.

Title: ICICS-PCM 2003 : proceedings of the 2003 joint conference of the 4th International conference on information, communications & signal processing and 4th Pacific-Rim conference on multimedia : 15-18 December 2003, Singapore.

Description: 3 v.

Copyright Claimant: IEEE

Date of Creation: 2003

Date of Publication: 2003-12-15

Other Title: 4th International conference on information, communications & signal processing
4th Pacific-Rim conference on multimedia
2003 joint conference of the 4th Int'l conf. on communications & signal processing
International conference on information, communications & signal processing
Pacific-Rim conference on multimedia

Names: [IEEE](#)

[◀ previous](#) [next ▶](#)

Save, Print and Email ([Help Page](#))

Select Download Format

Enter your email address:

[Help](#) [Search](#) [History](#) [Titles](#) [Start Over](#)

[Contact Us](#) | [Request Copies](#) | [Get a Search Estimate](#) | [Frequently Asked Questions \(FAQs\) about Copyright](#) | [Copyright Office Home Page](#) | [Library of Congress Home Page](#)

Ex. 1007E

Browse
searchJournal
SearchAll
DatabasesCitation
Linker

Sign in

Menu ▼



CONFERENCE PROCEEDING

ICICS-PCM 2003 : 2003 Joint Conference of the Fourth International Conference on Information, Communications, & Signal Processing and Fourth Pacific-Rim Conference on Multimedia : 15-18 December 2003, Meritus Mandarin Singapore Hotel, Singapore / [organised by School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore ; technical sponsorships by IEEE Singapore Communications Chapter, IEEE Singapore Signal Processing Chapter]

Joint Conference of the Fourth International Conference on Information, Communications, and Signal Processing and Fourth Pacific-Rim Conference on Multimedia (4th : 2003 : Singapore); Nanyang Technological University. School of Electrical and Electronic Engineering; IEEE Singapore Section. Communications Chapter; IEEE Singapore Section. Signal Processing Chapter; International Conference on Information, Communications, and Signal Processing; Pacific-Rim Conference on Multimedia.

c2003

Available at [Linda Hall Library Books - 2nd Floor \(TK5102.9 .I62 2003\)](#) >

Top

Send to

Get it

Details

Virtual Browse

Links

Send to



Email



QR



Permalink



Citation



EasyBib



Export RIS



Export to Excel

Get It

[Back to top of page](#)

LOCATION ITEMS

Linda Hall Library

Available , Books - 2nd Floor ; TK5102.9 .I62 2003



Item in place (0 requests)

Loanable



Item in place (0 requests)

Loanable



Item in place (0 requests)

Loanable



Details

Title	ICICS-PCM 2003 : 2003 Joint Conference of the Fourth International Conference on Information, Communications, & Signal Processing and Fourth Pacific-Rim Conference on Multimedia : 15-18 December 2003, Meritus Mandarin Singapore Hotel, Singapore / [organised by School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore ; technical sponsorships by IEEE Singapore Communications Chapter, IEEE Singapore Signal Processing Chapter]
Other title	Joint Conference of the Fourth International Conference on Information, Communications, and Signal Processing and Fourth Pacific-Rim on Multimedia
Creator/Contributor	Joint Conference of the Fourth International Conference on Information, Communications, and Signal Processing and Fourth Pacific-Rim Conference on Multimedia (4th : 2003 : Singapore) > Nanyang Technological University. School of Electrical and Electronic Engineering > IEEE Singapore Section. Communications Chapter > IEEE Singapore Section. Signal Processing Chapter > International Conference on Information, Communications, and Signal Processing > Pacific-Rim Conference on Multimedia. >
Subjects	Signal processing > Wireless communication systems > Multimedia systems > Information technology >
Identifier	ISBN : 9780780381858 ISBN : 0780381858 OCLC : (OCoLC)55661955
Publisher	New York : Institute of Electrical and Electronics Engineers
Creation date	c2003
Format	3 v. : ill. ; 30 cm.
General notes	"IEEE catalog number: 03EX758" --T.p. verso. Includes bibliographical references and indexes.

Virtual Browse





Links

[Display Source Record](#)

Linda Hall Library MARC record for ICICS-PCM 2003 at

https://catalog.lindahall.org/discovery/sourceRecord?vid=01LINDAHALL_INST:LHL&docId=alma993911243405961&recordOwner=01LINDAHALL_INST copied below.

leader 02058nam a2200361la 4500

001 993911243405961

005 20040616145300.0

008 040616s2003 nyua b 101 0 eng d

020 ##\$a9780780381858

020 ##\$a0780381858

035 ##\$a(OCOLC)55661955

035 ##\$9539874

035 ##\$a(MoKL)391124-lhalldb

035 ##\$a(lhalldb)391124-lhalldb

040 ##\$aLHL \$cLHL

049 ##\$aLHLA

090 ##\$aTK5102.9 \$b.l62 2003

111 2#\$aJoint Conference of the Fourth International Conference on Information, Communications, and Signal Processing and Fourth Pacific-Rim Conference on Multimedia \$n(4th : \$d2003 : \$cSingapore)

245 10\$aICICS-PCM 2003 : \$b2003 Joint Conference of the Fourth International Conference on Information, Communications, & Signal Processing and Fourth Pacific-Rim Conference on Multimedia : 15-18 December 2003, Meritus Mandarin Singapore Hotel, Singapore / \$c[organised by School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore ; technical sponsorships by IEEE Singapore Communications Chapter, IEEE Singapore Signal Processing Chapter]

246 3#\$aJoint Conference of the Fourth International Conference on Information, Communications, and Signal Processing and Fourth Pacific-Rim on Multimedia

260 ##\$a[New York] : \$bInstitute of Electrical and Electronics Engineers, \$cc2003.

300 ##\$a3 v. : \$bill. ; \$c30 cm.

500 ##\$a"IEEE catalog number: 03EX758"--T.p. verso.

504 ##\$aIncludes bibliographical references and indexes.

650 #0\$aSignal processing \$vCongresses

650 #0\$aWireless communication systems \$vCongresses

650 #0\$aMultimedia systems \$vCongresses

650 #0\$aInformation technology \$vCongresses

710 2#\$aNanyang Technological University. \$bSchool of Electrical and Electronic Engineering

710 2#\$aIEEE Singapore Section. \$bCommunications Chapter

710 2#\$aIEEE Singapore Section. \$bSignal Processing Chapter.

711 2#\$aInternational Conference on Information, Communications, and Signal Processing

711 2#\$aPacific-Rim Conference on Multimedia.

948 ##\$aLTI 09/01/2008

994 ##\$aEO \$bLHL

Ex. 1007F

libraries.org

A directory of libraries throughout the world

- Home »
- Libraries »
- Guides
- Documents
- Vendors »
- Products »
- News
- Procurement
- Member Login »

found 2 items where the item contains Linda hall Library. Showing item 1 of 2

☐ list

☐ next >>

search:

Linda Hall Library

Linda Hall Library of Science, Engineering and Technology

Address: 5109 Cherry Street
Kansas City, Missouri
64110-2498
United States



County: Jackson

Phone: 816-363-4600

Connect to: Library Web Site ☒ Online Catalog

Library details: Linda Hall Library is a Independent Research library. This library is affiliated with Linda Hall Library (view map) . The collection of the library contains 860,000 volumes. The library circulates 30,000 items per year.

Other Info: The Linda Hall Library is the world's foremost independent research library devoted to science, engineering, and technology. A not-for-profit, privately funded institution, the Library is open to the public free of charge and does circulate. This library shares an ILS with the Spencer Art Reference Library (of the Nelson-Atkins Museum of Art, also in Kansas City, MO). The combined catalog was formerly called LEONARDO (named for the artist/inventor Leonardo da Vinci). LHL is a member of the Independent Research Libraries Association.

Permalink: https://librarytechnology.org/library/2294
(Use this link to refer back to this listing.)

Administration: The director of the library is Eric Dorfman .

Organizational structure: The library is part of a privately-funded organization. It operates on a Non-profit financial model.

Collection size	860,000	volumes
Annual Circulation	30,000	transactions

Technology Profile

	Product Name	Year Contracted
Current Automation System	Alma	2018



Linda Hall Library



map location

Related Libraries

- Libraries located in Kansas City (Missouri)
- Libraries located in Jackson county (Missouri)
- View map of libraries in Jackson County
- all Independent Research libs in Missouri

About libraries.org

libraries.org is a free directory of libraries throughout the world.

This directory is a component of Library Technology Guides.

Previous Automation System	Voyager	2010
Previous Automation System	Horizon	1994
Previous Automation System	None	
Discovery Service (w/index)	Ex Libris CDI	2010
Discovery Interface	Primo	2010
Digital Collection Management	CONTENTdm	
Web Content Management	WordPress	
Item ID Type	Barcode	
The library's automation system is hosted by the vendor through an Software-as-a-Service (SaaS) arrangement.		
This library is responsible for the procurement of the library automation system.		

Please contact [Marshall Breeding](#) to report corrections about information listed for this library.

Automation Survey: We are conducting a study on the automation systems used in libraries. This survey should be completed by a person familiar with your libraries use of **Alma**.

[Respond to the 2024 Library Automation Perceptions Survey](#)

Identifiers	
libraries.org ID	2294
OCLC Symbol	LHL
WorldCat Registry ID	99283

Record History: This listing was created on Aug 11, 1998 and was last modified on Dec 10, 2024.

Updates: Corrections or Updates? Registered members of Library Technology Guides can submit updates to library listings in libraries.org. **Registration** is free and easy. Already registered? [login](#). Or, you can report corrections just by sending a message to [Marshall Breeding](#).

[libraries.org](#) [Update this Entry](#) [Library Technology Guides](#)

Maintained by [Marshall Breeding](#)

terms of service

Copyright 1994-2024

A Guide to MPEG Fundamentals and Protocol Analysis

(Including DVB and ATSC)



Tektronix

A Guide to MPEG Fundamentals and Protocol Analysis

(Including DVB and ATSC)



A Guide to MPEG Fundamentals and Protocol Analysis

▶ **A Guide to MPEG Fundamentals and Protocol Analysis**

▶ **Contents**

▶ Section 1 - Introduction to MPEG	1
1.1 Convergence	1
1.2 Why Compression is Needed	1
1.3 Applications of Compression	1
1.4 Introduction to Video Compression	2
1.5 Introduction to Audio Compression	4
1.6 MPEG Signals	4
1.7 Need for Monitoring and Analysis	5
1.8 Pitfalls of Compression	6
▶ Section 2 - Compression in Video	7
2.1 Spatial or Temporal Coding?	7
2.2 Spatial Coding	7
2.3 Weighting	9
2.4 Scanning	10
2.5 Entropy Coding	10
2.6 A Spatial Coder	11
2.7 Temporal Coding	12
2.8 Motion Compensation	13
2.9 Bidirectional Coding	15
2.10 I, P, and B Pictures	15
2.11 An MPEG Compressor	17
2.12 Preprocessing	20
2.13 Profiles and Levels	21
2.14 Wavelets	22
▶ Section 3 - Audio Compression	23
3.1 The Hearing Mechanism	23
3.2 Subband Coding	24
3.3 MPEG Layer 1	25
3.4 MPEG Layer 2	26
3.5 Transform Coding	26
3.6 MPEG Layer 3	27
3.7 AC-3	27
▶ Section 4 - Elementary Streams	29
4.1 Video Elementary Stream Syntax	29
4.2 Audio Elementary Streams	30

▶ Section 5 – Packetized Elementary Streams (PES)	31
5.1 PES Packets	31
5.2 Time Stamps	31
5.3 PTS/DTS	31
▶ Section 6 – Program Streams	33
6.1 Recording vs. Transmission	33
6.2 Introduction to Program Streams	33
▶ Section 7 – Transport Streams	35
7.1 The Job of a Transport Stream	35
7.2 Packets	36
7.3 Program Clock Reference (PCR)	36
7.4 Packet Identification (PID)	37
7.5 Program Specific Information (PSI)	37
▶ Section 8 – Introduction to DVB/ATSC	39
8.1 An Overall View	39
8.2 Remultiplexing	40
8.3 Service Information (SI)	40
8.4 Error Correction	40
8.5 Channel Coding	41
8.6 Inner Coding	43
8.7 Transmitting Digits	44
▶ Section 9 – MPEG Testing	45
9.1 Testing Requirements	45
9.2 Analyzing a Transport Stream	45
9.3 Hierarchic View	46
9.4 Interpreted View	48
9.5 Syntax and CRC Analysis	48
9.6 Filtering	49
9.7 Timing Analysis	49
9.8 Elementary Stream Testing	51
9.9 Sarnoff Compliant Bit Streams	51
9.10 Elementary Stream Analysis	52
9.11 Creating a Transport Stream	52
9.12 Jitter Generation	53
9.13 DVB Tests	53
▶ Glossary	55

► Section 1 - Introduction to MPEG

MPEG is one of the most popular audio/video compression techniques because it is not just a single standard. Instead, it is a range of standards suitable for different applications but based on similar principles. MPEG is an acronym for the Moving Picture Experts Group which was set up by the ISO (International Standards Organization) to work on compression.

MPEG can be described as the interaction of acronyms. As ETSI stated "The CAT is a pointer to enable the IRD to find the EMMs associated with the CA system(s) that it uses." If you can understand that sentence you don't need this book!

1.1 Convergence

Digital techniques have made rapid progress in audio and video for a number of reasons. Digital information is more robust and can be coded to substantially eliminate error. This means that generation-loss in recording and losses in transmission are eliminated. The Compact Disc was the first consumer product to demonstrate this.

While the CD has an improved sound quality with respect to its vinyl predecessor, comparison of quality alone misses the point. The real point is that digital recording and transmission techniques allow content manipulation to a degree that is impossible with analog. Once audio or video are digitized, they become data. Such data cannot be distinguished from any other kind of data; therefore, digital video and audio become the province of computer technology.

The convergence of computers and audio/video is an inevitable consequence of the key inventions of computing and Pulse Code Modulation. Digital media can store any type of information, so it is easy to use a computer storage device for digital video. The nonlinear workstation was the first example of an application of convergent technology that did not have an analog forerunner. Another example, multimedia, mixed the storage of audio, video, graphics, text, and data on the same medium. Multimedia is impossible in the analog domain.

1.2 Why Compression is Needed

The initial success of digital video was in post-production applications, where the high cost of digital video was offset by its limitless layering and effects capability. However, production-standard digital video generates over 200 megabits per second of data and this bit rate requires extensive capacity for storage and wide bandwidth for transmission. Digital video could only be used in wider applications if the storage and bandwidth requirements could be eased; easing these requirements is the purpose of compression.

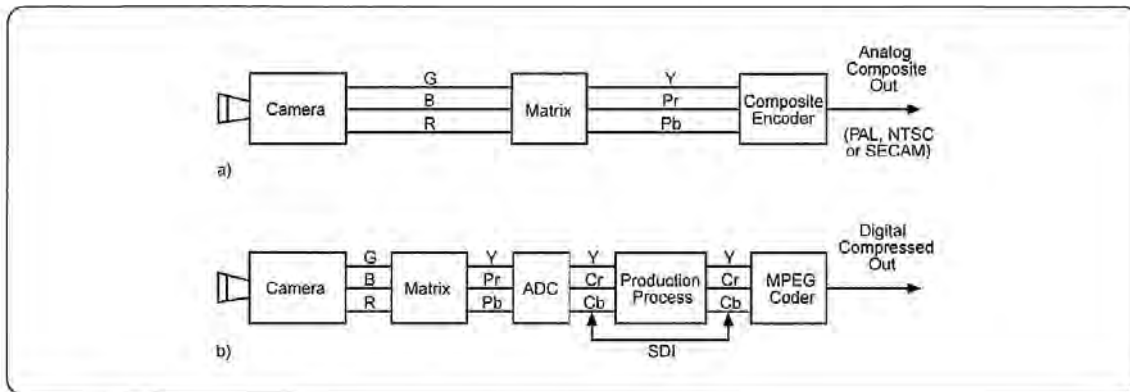
Compression is a way of expressing digital audio and video by using less data. Compression has the following advantages:

- A smaller amount of storage is needed for a given amount of source material. With high-density recording such as with tape, compression allows highly miniaturized equipment for consumer and Electronic News Gathering (ENG) use. The access time of tape improves with compression because less tape needs to be shuttled to skip over a given amount of program. With expensive storage media such as RAM, compression makes new applications affordable.
- When working in real time, compression reduces the bandwidth needed. Additionally, compression allows faster-than-real-time transfer between media, for example, between tape and disk.
- A compressed recording format can afford a lower recording density and this can make the recorder less sensitive to environmental factors and maintenance.

1.3 Applications of Compression

Compression has a long association with television. Interlace is a simple form of compression giving a 2:1 reduction in bandwidth. The use of color-difference signals instead of GBR is another form of compression. Because the eye is less sensitive to color detail, the color-difference signals need less bandwidth. When color broadcasting was introduced, the channel structure of monochrome had to be retained and composite video was developed. Composite video systems, such as PAL, NTSC, and SECAM, are forms of compression because they use the same bandwidth for color as was used for monochrome.

A Guide to MPEG Fundamentals and Protocol Analysis



► Figure 1-1.

Figure 1-1a shows that in traditional television systems, the GBR camera signal is converted to Y, Pr, Pb components for production and encoded into analog composite for transmission. Figure 1-1b shows the modern equivalent. The Y, Pr, Pb signals are digitized and carried as Y, Cr, Cb signals in SDI form through the production process prior to being encoded with MPEG for transmission. Clearly, MPEG can be considered by the broadcaster as a more efficient replacement for composite video. In addition, MPEG has greater flexibility because the bit rate required can be adjusted to suit the application. At lower bit rates and resolutions, MPEG can be used for video conferencing and video telephones.

DVB and ATSC (the European- and American-originated digital-television broadcasting standards) would not be viable without compression because the bandwidth required would be too great. Compression extends the playing time of DVD (digital video/versatile disc) allowing full-length movies on a standard size compact disc. Compression also reduces the cost of Electronic News Gathering and other contributions to television production.

In tape recording, mild compression eases tolerances and adds reliability in Digital Betacam and Digital-S, whereas in SX, DVC, DVCPRC, and DVCAM, the goal is miniaturization. In magnetic disk drives, such as the Tektronix Profile® storage system, that are used in file servers and networks (especially for news purposes), compression lowers storage cost. Compression also lowers bandwidth, which allows more users to access a given server. This characteristic is also important for VOD (Video On Demand) applications.

1.4 Introduction to Video Compression

In all real program material, there are two types of components of the signal: those which are novel and unpredictable and those which can be anticipated. The

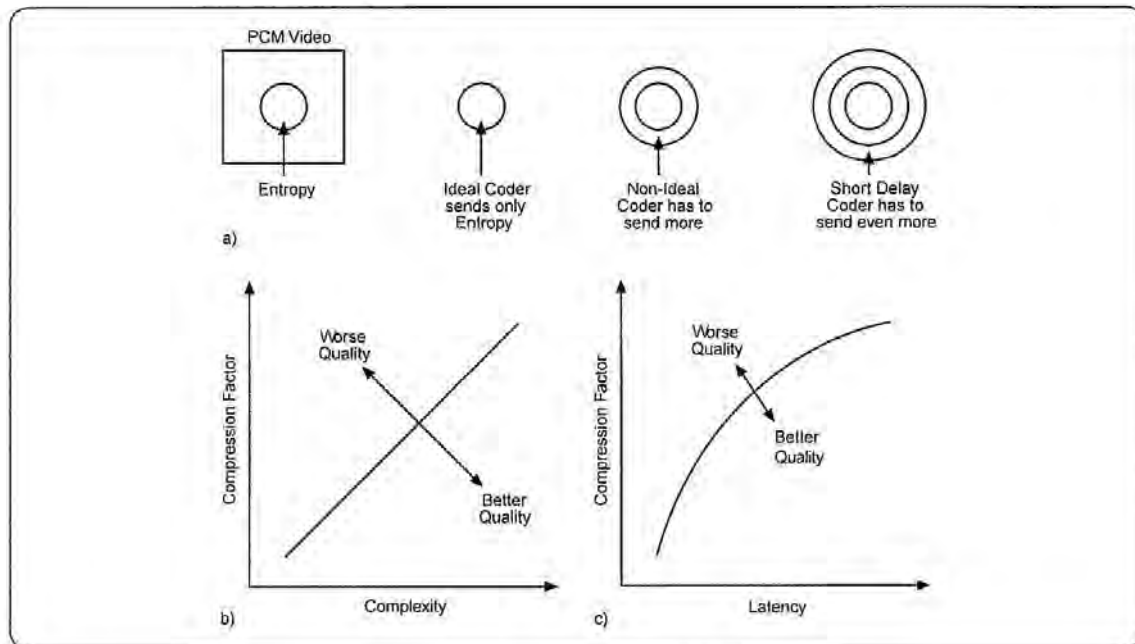
novel component is called entropy and is the true information in the signal. The remainder is called redundancy because it is not essential. Redundancy may be spatial, as it is in large plain areas of picture where adjacent pixels have almost the same value. Redundancy can also be temporal as it is where similarities between successive pictures are used. All compression systems work by separating the entropy from the redundancy in the encoder. Only the entropy is recorded or transmitted and the decoder computes the redundancy from the transmitted signal. Figure 1-2a shows this concept.

An ideal encoder would extract all the entropy and only this will be transmitted to the decoder. An ideal decoder would then reproduce the original signal. In practice, this ideal cannot be reached. An ideal coder would be complex and cause a very long delay in order to use temporal redundancy. In certain applications, such as recording or broadcasting, some delay is acceptable, but in videoconferencing it is not. In some cases, a very complex coder would be too expensive. It follows that there is no one ideal compression system.

In practice, a range of coders is needed which have a range of processing delays and complexities. The power of MPEG is that it is not a single compression format, but a range of standardized coding tools that can be combined flexibly to suit a range of applications. The way in which coding has been performed is included in the compressed data so that the decoder can automatically handle whatever the coder decided to do.

MPEG coding is divided into several profiles that have different complexity, and each profile can be implemented at a different level depending on the resolution of the input picture. Section 2 considers profiles and levels in detail.

② www.tektronix.com/video_audio/



► Figure 1-2.

There are many different digital video formats and each has a different bit rate. For example a high definition system might have six times the bit rate of a standard definition system. Consequently, just knowing the bit rate out of the coder is not very useful. What matters is the compression factor, which is the ratio of the input bit rate to the compressed bit rate, for example 2:1, 5:1, and so on.

Unfortunately the number of variables involved make it very difficult to determine a suitable compression factor. Figure 1-2a shows that for an ideal coder, if all of the entropy is sent, the quality is good. However, if the compression factor is increased in order to reduce the bit rate, not all of the entropy is sent and the quality falls. Note that in a compressed system when the quality loss occurs, it is steep (Figure 1-2b). If the available bit rate is inadequate, it is better to avoid this area by reducing the entropy of the input picture. This can be done by filtering. The loss of resolution caused by the filtering is subjectively more acceptable than the compression artifacts.

To identify the entropy perfectly, an ideal compressor would have to be extremely complex. A practical compressor may be less complex for economic

reasons and must send more data to be sure of carrying all of the entropy. Figure 1-2b shows the relationship between coder complexity and performance. The higher the compression factor required, the more complex the encoder has to be.

The entropy in video signals varies. A recording of an announcer delivering the news has much redundancy and is easy to compress. In contrast, it is more difficult to compress a recording with leaves blowing in the wind or one of a football crowd that is constantly moving and therefore has less redundancy (more information or entropy). In either case, if all the entropy is not sent, there will be quality loss. Thus, we may choose between a constant bit-rate channel with variable quality or a constant quality channel with variable bit rate. Telecommunications network operators tend to prefer a constant bit rate for practical purposes, but a buffer memory can be used to average out entropy variations if the resulting increase in delay is acceptable. In recording, a variable bit rate may be easier to handle and DVD uses variable bit rate, speeding up the disc where difficult material exists.

Intra-coding (intra = within) is a technique that exploits spatial redundancy, or redundancy within the picture; inter-coding (inter = between) is a technique

A Guide to MPEG Fundamentals and Protocol Analysis

that exploits temporal redundancy. Intra-coding may be used alone, as in the JPEG standard for still pictures, or combined with inter-coding as in MPEG.

Intra-coding relies on two characteristics of typical images. First, not all spatial frequencies are simultaneously present, and second, the higher the spatial frequency, the lower the amplitude is likely to be. Intra-coding requires analysis of the spatial frequencies in an image. This analysis is the purpose of transforms such as wavelets and DCT (discrete cosine transform). Transforms produce coefficients which describe the magnitude of each spatial frequency. Typically, many coefficients will be zero, or nearly zero, and these coefficients can be omitted, resulting in a reduction in bit rate.

Inter-coding relies on finding similarities between successive pictures. If a given picture is available at the decoder, the next picture can be created by sending only the picture differences. The picture differences will be increased when objects move, but this magnification can be offset by using motion compensation, since a moving object does not generally change its appearance very much from one picture to the next. If the motion can be measured, a closer approximation to the current picture can be created by shifting part of the previous picture to a new location. The shifting process is controlled by a vector that is transmitted to the decoder. The vector transmission requires less data than sending the picture-difference data.

MPEG can handle both interlaced and non-interlaced images. An image at some point on the time axis is called a "picture," whether it is a field or a frame.

Interlace is not ideal as a source for digital compression because it is in itself a compression technique. Temporal coding is made more complex because pixels in one field are in a different position to those in the next.

Motion compensation minimizes but does not eliminate the differences between successive pictures. The picture-difference is itself a spatial image and can be compressed using transform-based intra-coding as previously described. Motion compensation simply reduces the amount of data in the difference image.

The efficiency of a temporal coder rises with the time span over which it can act. Figure 1-2c shows that if a high compression factor is required, a longer time span in the input must be considered and thus a longer coding delay will be experienced. Clearly, temporally coded signals are difficult to edit because the content of a given output picture may be based on image data which was trans-

mitted some time earlier. Production systems will have to limit the degree of temporal coding to allow editing and this limitation will in turn limit the available compression factor.

1.5 Introduction to Audio Compression

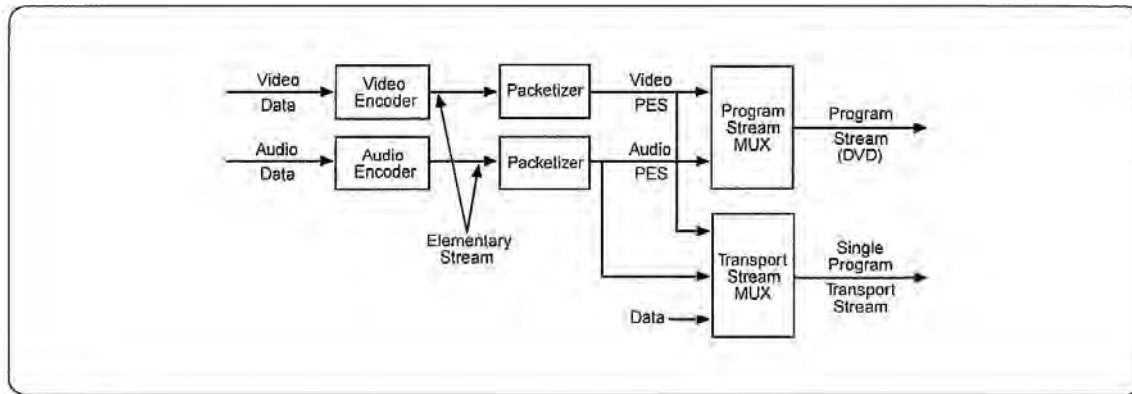
The bit rate of a PCM digital audio channel is only about one megabit per second, which is about 0.5% of 4:2:2 digital video. With mild video compression schemes, such as Digital Betacam, audio compression is unnecessary. But, as the video compression factor is raised, it becomes necessary to compress the audio as well.

Audio compression takes advantage of two facts. First, in typical audio signals, not all frequencies are simultaneously present. Second, because of the phenomenon of masking, human hearing cannot discern every detail of an audio signal. Audio compression splits the audio spectrum into bands by filtering or transforms, and includes less data when describing bands in which the level is low. Where masking prevents or reduces audibility of a particular band, even less data needs to be sent.

Audio compression is not as easy to achieve as is video compression because of the acuity of hearing. Masking only works properly when the masking and the masked sounds coincide spatially. Spatial coincidence is always the case in mono recordings but not in stereo recordings, where low-level signals can still be heard if they are in a different part of the sound stage. Consequently, in stereo and surround sound systems, a lower compression factor is allowable for a given quality. Another factor complicating audio compression is that delayed resonances in poor loudspeakers actually mask compression artifacts. Testing a compressor with poor speakers gives a false result, and signals which are apparently satisfactory may be disappointing when heard on good equipment.

1.6 MPEG Signals

The output of a single MPEG audio or video coder is called an Elementary Stream. An Elementary Stream is an endless near real-time signal. For convenience, it can be broken into convenient-sized data blocks in a Packetized Elementary Stream (PES). These data blocks need header information to identify the start of the packets and must include time stamps because packetizing disrupts the time axis.



► Figure 1-3.

Figure 1-3 shows that one video PES and a number of audio PES can be combined to form a Program Stream, provided that all of the coders are locked to a common clock. Time stamps in each PES ensure lip-sync between the video and audio. Program Streams have variable-length packets with headers. They find use in data transfers to and from optical and hard disks, which are error free and in which files of arbitrary sizes are expected. DVD uses Program Streams.

For transmission and digital broadcasting, several programs and their associated PES can be multiplexed into a single Transport Stream. A Transport Stream differs from a Program Stream in that the PES packets are further subdivided into short fixed-size packets and in that multiple programs encoded with different clocks can be carried. This is possible because a transport stream has a program clock reference (PCR) mechanism which allows transmission of multiple clocks, one of which is selected and regenerated at the decoder. A Single Program Transport Stream (SPTS) is also possible and this may be found between a coder and a multiplexer. Since a Transport Stream can genlock the decoder clock to the encoder clock, the Single Program Transport Stream (SPTS) is more common than the Program Stream.

A Transport Stream is more than just a multiplex of audio and video PES. In addition to the compressed audio, video, and data, a Transport Stream includes a great deal of metadata describing the bit stream. This includes the Program Association Table (PAT) that lists every program in the transport stream. Each entry in the PAT points to a Program Map Table (PMT) that lists the elementary streams making up each program. Some programs will be open, but some programs may be subject to conditional access (encryption) and this information is also carried in the metadata.

The Transport Stream consists of fixed-size data packets, each containing 188 bytes. Each packet carries a packet identifier code (PID). Packets in the same elementary stream all have the same PID, so that the decoder (or a demultiplexer) can select the elementary stream(s) it wants and reject the remainder. Packet-continuity counts ensure that every packet that is needed to decode a stream is received. An effective synchronization system is needed so that decoders can correctly identify the beginning of each packet and deserialize the bit stream into words.

1.7 Need for Monitoring and Analysis

The MPEG transport stream is an extremely complex structure using interlinked tables and coded identifiers to separate the programs and the elementary streams within the programs. Within each elementary stream, there is a complex structure, allowing a decoder to distinguish between, for example, vectors, coefficients and quantization tables.

Failures can be divided into two broad categories. In the first category, the transport system correctly multiplexes and delivers information from an encoder to a decoder with no bit errors or added jitter, but the encoder or the decoder has a fault. In the second category, the encoder and decoder are fine, but the transport of data from one to the other is defective. It is very important to know whether the fault lies in the encoder, the transport, or the decoder if a prompt solution is to be found.

Synchronizing problems, such as loss or corruption of sync patterns, may prevent reception of the entire transport stream. Transport-stream protocol defects may prevent the decoder from finding all of the data for a program, perhaps

A Guide to MPEG Fundamentals and Protocol Analysis

delivering picture but not sound. Correct delivery of the data but with excessive jitter can cause decoder timing problems.

If a system using an MPEG transport stream fails, the fault could be in the encoder, the multiplexer, or in the decoder. How can this fault be isolated? First, verify that a transport stream is compliant with the MPEG-coding standards. If the stream is not compliant, a decoder can hardly be blamed for having difficulty. If it is, the decoder may need attention.

Traditional video testing tools, the signal generator, the waveform monitor, and vectorscope, are not appropriate in analyzing MPEG systems, except to ensure that the video signals entering and leaving an MPEG system are of suitable quality. Instead, a reliable source of valid MPEG test signals is essential for testing receiving equipment and decoders. With a suitable analyzer, the performance of encoders, transmission systems, multiplexers, and remultiplexers can be assessed with a high degree of confidence. As a long standing supplier of high grade test equipment to the video industry, Tektronix continues to provide test and measurement solutions as the technology evolves, giving the MPEG user the confidence that complex compressed systems are correctly functioning and allowing rapid diagnosis when they are not.

1.8 Pitfalls of Compression

MPEG compression is lossy in that what is decoded, is not identical to the original. The entropy of the source varies, and when entropy is high, the compression system may leave visible artifacts when decoded. In temporal compression, redundancy between successive pictures is assumed. When this is not the case, the system fails. An example is video from a press conference where flashguns are firing. Individual pictures containing the flash are totally different from their neighbors, and coding artifacts become obvious.

Irregular motion or several independently moving objects on screen require a lot of vector bandwidth and this requirement may only be met by reducing the picture-data bandwidth. Again, visible artifacts may occur whose level varies and depends on the motion. This problem often occurs in sports-coverage video.

Coarse quantizing results in luminance contouring and posterized color. These can be seen as blotchy shadows and blocking on large areas of plain color. Subjectively, compression artifacts are more annoying than the relatively constant impairments resulting from analog television transmission systems.

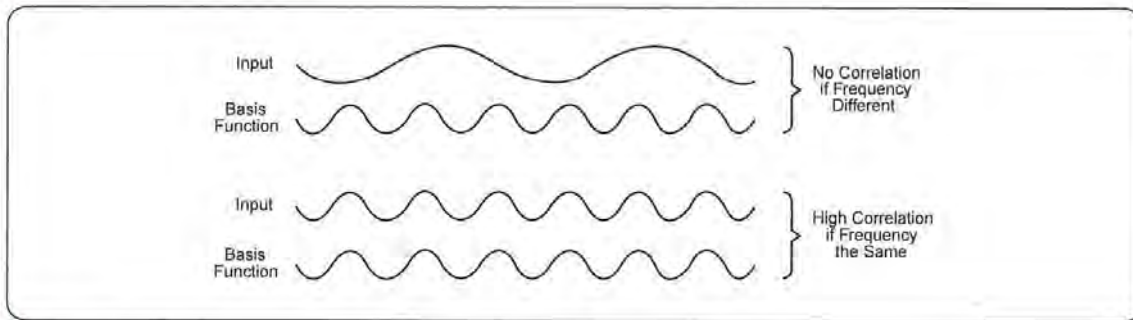
The only solution to these problems is to reduce the compression factor. Consequently, the compression user has to make a value judgment between the economy of a high compression factor and the level of artifacts.

In addition to extending the encoding and decoding delay, temporal coding also causes difficulty in editing. In fact, an MPEG bit stream cannot be arbitrarily edited at all. This restriction occurs because, in temporal coding, the decoding of one picture may require the contents of an earlier picture and the contents may not be available following an edit. The fact that pictures may be sent out of sequence also complicates editing.

If suitable coding has been used, edits can take place only at splice points, which are relatively widely spaced. If arbitrary editing is required, the MPEG stream must undergo a read-modify-write process, which will result in generation loss.

The viewer is not interested in editing, but the production user will have to make another value judgment about the edit flexibility required. If greater flexibility is required, the temporal compression has to be reduced and a higher bit rate will be needed.

► Section 2 - Compression in Video



► Figure 2-1.

This section shows how video compression is based on the perception of the eye. Important enabling techniques, such as transforms and motion compensation, are considered as an introduction to the structure of an MPEG coder.

2.1 Spatial or Temporal Coding?

As was seen in Section 1, video compression can take advantage of both spatial and temporal redundancy. In MPEG, temporal redundancy is reduced first by using similarities between successive pictures. As much as possible of the current picture is created or "predicted" by using information from pictures already sent. When this technique is used, it is only necessary to send a difference picture, which eliminates the differences between the actual picture and the prediction. The difference picture is then subject to spatial compression. As a practical matter it is easier to explain spatial compression prior to explaining temporal compression.

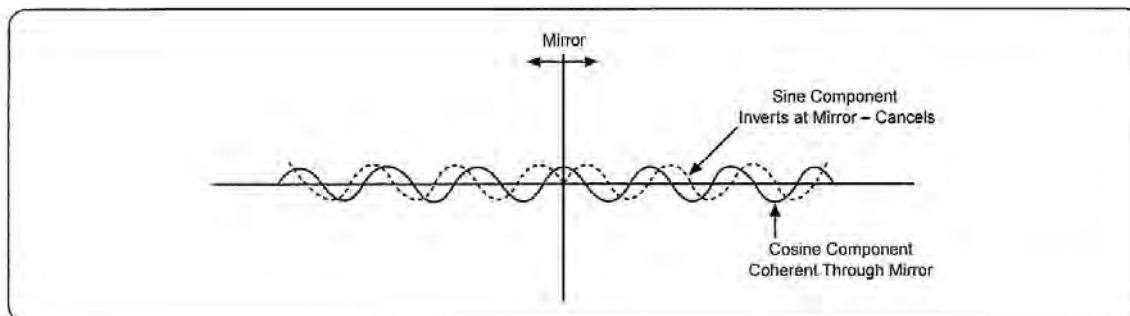
Spatial compression relies on similarities between adjacent pixels in plain areas of picture and on dominant spatial frequencies in areas of patterning. The JPEG system uses spatial compression only, since it is designed to transmit individual still pictures. However, JPEG may be used to code a succession of individual pictures for video. In the so-called "Motion JPEG" application, the compression factor will not be as good as if temporal coding was used, but the bit stream will be freely editable on a picture-by-picture basis.

2.2 Spatial Coding

The first step in spatial coding is to perform an analysis of spatial frequency using a transform. A transform is simply a way of expressing a waveform in a different domain, in this case, the frequency domain. The output of a transform is a set of coefficients that describe how much of a given frequency is present. An inverse transform reproduces the original waveform. If the coefficients are handled with sufficient accuracy, the output of the inverse transform is identical to the original waveform.

The most well known transform is the Fourier transform. This transform finds each frequency in the input signal. It finds each frequency by multiplying the input waveform by a sample of a target frequency, called a basis function, and integrating the product. Figure 2-1 shows that when the input waveform does not contain the target frequency, the integral will be zero, but when it does, the integral will be a coefficient describing the amplitude of that component frequency.

The results will be as described if the frequency component is in phase with the basis function. However if the frequency component is in quadrature with the basis function, the integral will still be zero. Therefore, it is necessary to perform two searches for each frequency, with the basis functions in quadrature with one another so that every phase of the input will be detected.



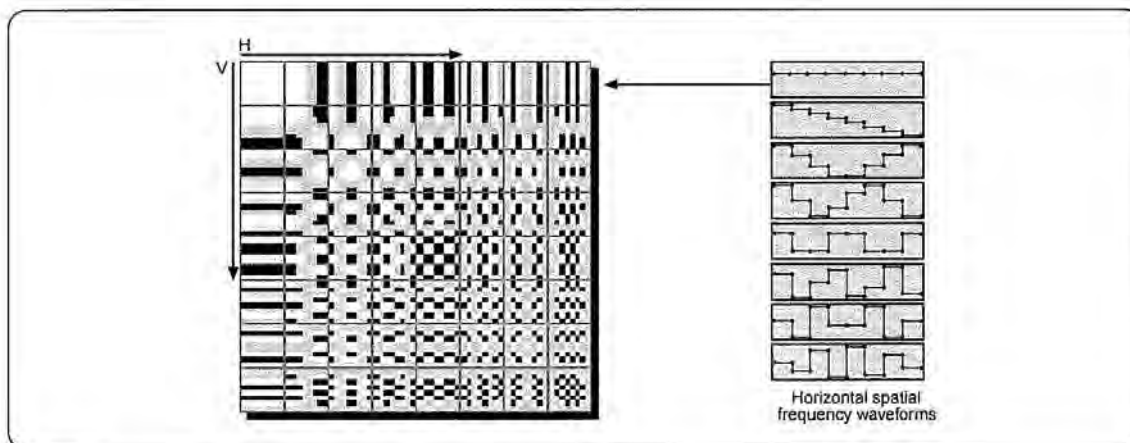
► Figure 2-2.

The Fourier transform has the disadvantage of requiring coefficients for both sine and cosine components of each frequency. In the cosine transform, the input waveform is time-mirrored with itself prior to multiplication by the basis functions. Figure 2-2 shows that this mirroring cancels out all sine components and doubles all of the cosine components. The sine basis function is unnecessary and only one coefficient is needed for each frequency.

The discrete cosine transform (DCT) is the sampled version of the cosine transform and is used extensively in two-dimensional form in MPEG. A block of 8 x 8 pixels is transformed to become a block of 8 x 8 coefficients. Since the transform requires multiplication by fractions, there is wordlength extension, resulting in coefficients that have longer wordlength than the pixel values.

Typically an 8-bit pixel block results in an 11-bit coefficient block. Thus, a DCT does not result in any compression; in fact it results in the opposite. However, the DCT converts the source pixels into a form where compression is easier.

Figure 2-3 shows the results of an inverse transform of each of the individual coefficients of an 8 x 8 DCT. In the case of the luminance signal, the top-left coefficient is the average brightness or DC component of the whole block. Moving across the top row, horizontal spatial frequency increases. Moving down the left column, vertical spatial frequency increases. In real pictures, different vertical and horizontal spatial frequencies may occur simultaneously and a coefficient at some point within the block will represent all possible horizontal and vertical combinations.



► Figure 2-3.

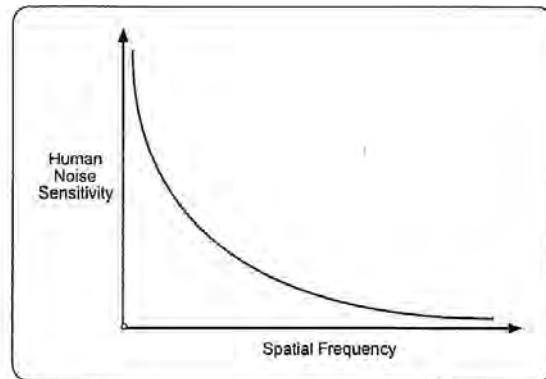
Figure 2-3 also shows the coefficients as a one-dimensional horizontal waveform. Combining these waveforms with various amplitudes and either polarity can reproduce any combination of 8 pixels. Thus combining the 64 coefficients of the 2-D DCT will result in the original 8 x 8 pixel block. Clearly for color pictures, the color difference samples will also need to be handled. Y, Cr, and Cb data are assembled into separate 8 x 8 arrays and are transformed individually. In much real program material, many of the coefficients will have zero or near-zero values and, therefore, will not be transmitted. This fact results in significant compression that is virtually lossless. If a higher compression factor is needed, then the wordlength of the non-zero coefficients must be reduced. This reduction will reduce accuracy of these coefficients and will introduce losses into the process. With care, the losses can be introduced in a way that is least visible to the viewer.

2.3 Weighting

Figure 2-4 shows that the human perception of noise in pictures is not uniform but is a function of the spatial frequency. More noise can be tolerated at high spatial frequencies. Also, video noise is effectively masked by fine detail in the picture, whereas in plain areas it is highly visible. The reader will be aware that traditional noise measurements are always weighted so that the technical measurement relates to the subjective result.

Compression reduces the accuracy of coefficients and has a similar effect to using shorter wordlength samples in PCM; that is, the noise level rises. In PCM, the result of shortening the word-length is that the noise level rises equally at all frequencies. As the DCT splits the signal into different frequencies, it becomes possible to control the spectrum of the noise. Effectively, low-frequency coefficients are rendered more accurately than high-frequency coefficients by a process of weighting.

Figure 2-5 shows that, in the weighting process, the coefficients from the DCT are divided by constants that are a function of two-dimensional frequency. Low-frequency coefficients will be divided by small numbers, and high-frequency coefficients will be divided by large numbers. Following the division, the least-significant bit is discarded or truncated. This truncation is a form of requantizing. In the absence of weighting, this requantizing would have the effect of doubling the size of the quantizing step, but with weighting, it increases the step size according to the division factor.



► Figure 2-4.

As a result, coefficients representing low spatial frequencies are requantized with relatively small steps and suffer little increased noise. Coefficients representing higher spatial frequencies are requantized with large steps and suffer more noise. However, fewer steps means that fewer bits are needed to identify the step and a compression is obtained.

In the decoder, a low-order zero will be added to return the weighted coefficients to their correct magnitude. They will then be multiplied by inverse weighting factors. Clearly, at high frequencies the multiplication factors will be larger, so the requantizing noise will be greater. Following inverse weighting, the coefficients will have their original DCT output values, plus requantizing error, which will be greater at high frequency than at low frequency.

As an alternative to truncation, weighted coefficients may be nonlinearly requantized so that the quantizing step size increases with the magnitude of the coefficient. This technique allows higher compression factors but worse levels of artifacts.

Clearly, the degree of compression obtained and, in turn, the output bit rate obtained, is a function of the severity of the requantizing process. Different bit rates will require different weighting tables. In MPEG, it is possible to use various different weighting tables and the table in use can be transmitted to the decoder, so that correct decoding automatically occurs.

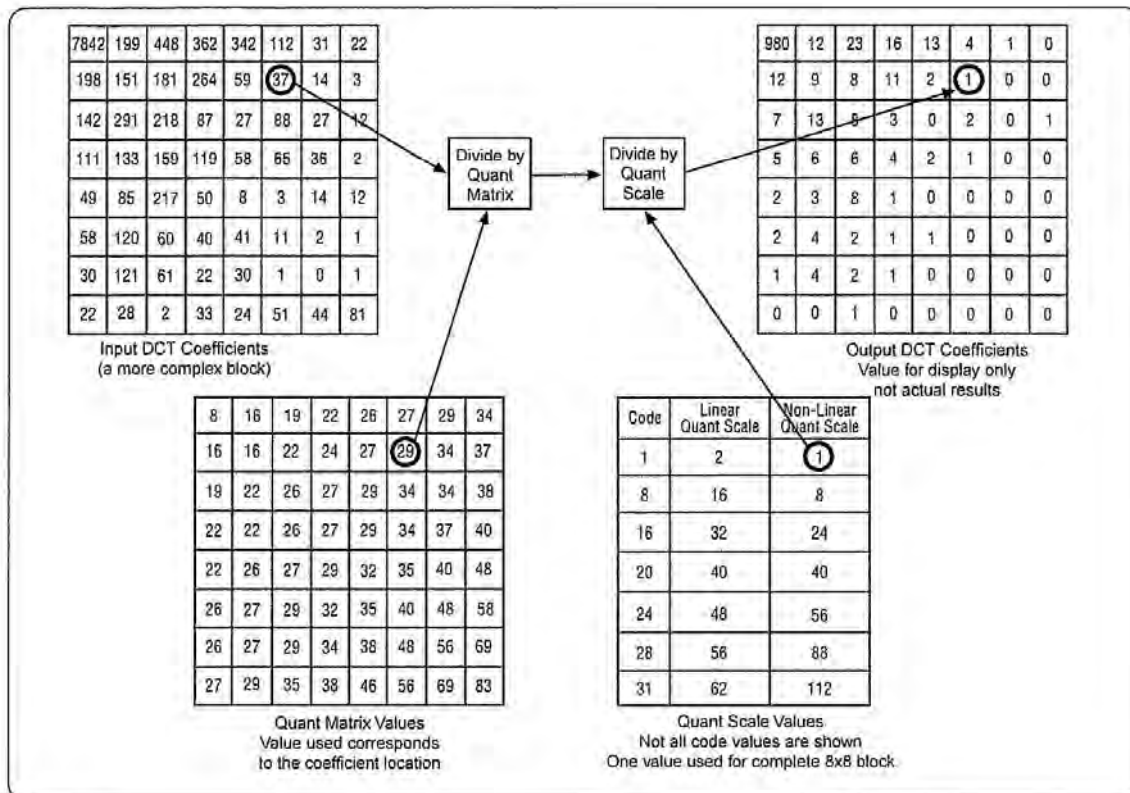


Figure 2-5.

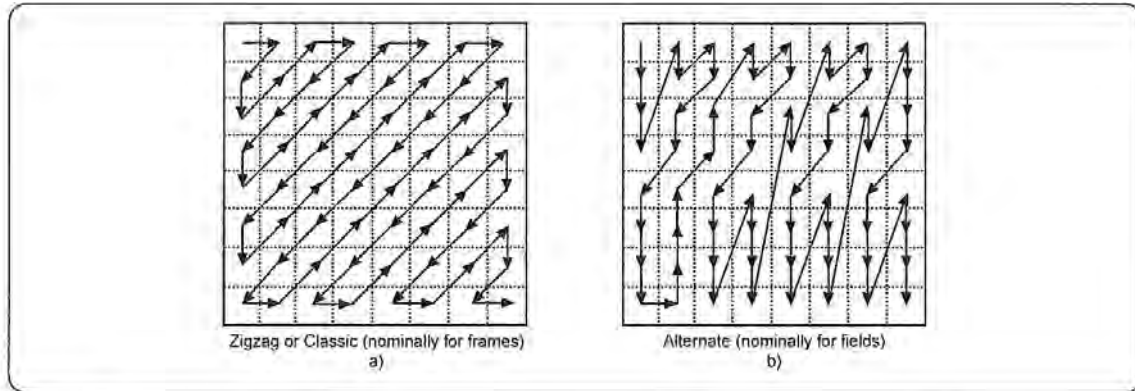
2.4 Scanning

In typical program material, the significant DCT coefficients are generally found in the top-left corner of the matrix. After weighting, low-value coefficients might be truncated to zero. More efficient transmission can be obtained if all of the non-zero coefficients are sent first, followed by a code indicating that the remainder are all zero. Scanning is a technique which increases the probability of achieving this result, because it sends coefficients in descending order of magnitude probability. Figure 2-6a shows that in a noninterlaced system, the probability of a coefficient having a high value is highest in the top-left corner and lowest in the bottom-right corner. A 45 degree diagonal zig-zag scan is the best sequence to use here.

In Figure 2-6b, the scan for an interlaced source is shown. In an interlaced picture, an 8 x 8 DCT block from one field extends over twice the vertical screen area, so that for a given picture detail, vertical frequencies will appear to be twice as great as horizontal frequencies. Thus, the ideal scan for an interlaced picture will be on a diagonal that is twice as steep. Figure 2-6b shows that a given vertical spatial frequency is scanned before scanning the same horizontal spatial frequency.

2.5 Entropy Coding

In real video, not all spatial frequencies are simultaneously present; therefore, the DCT coefficient matrix will have zero terms in it. Despite the use of scanning, zero coefficients will still appear between the significant values. Run-length coding (RLC) allows these coefficients to be handled more efficiently.



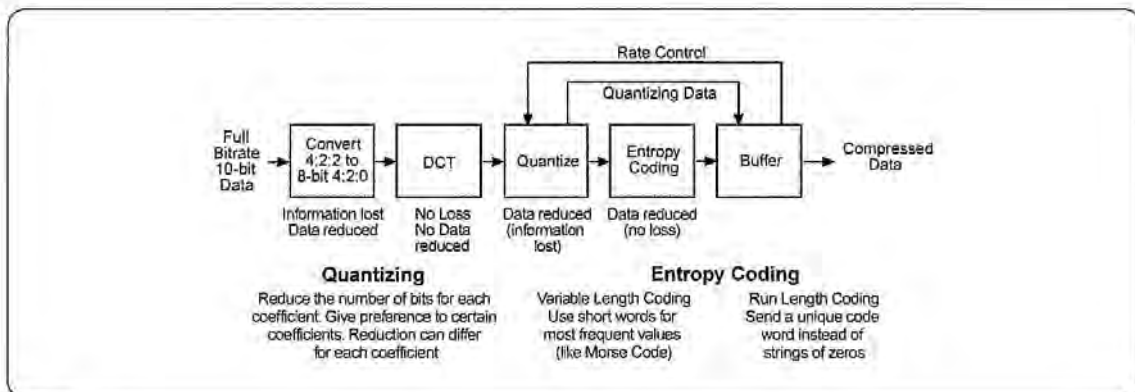
► Figure 2-6.

Where repeating values, such as a string of 0s, are present, run-length coding simply transmits the number of zeros rather than each individual bit.

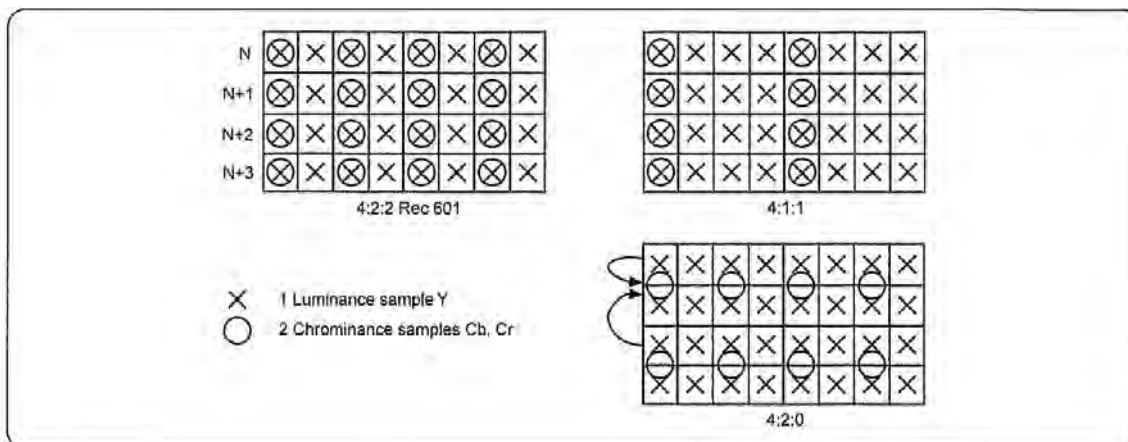
The probability of occurrence of particular coefficient values in real video can be studied. In practice, some values occur very often; others occur less often. This statistical information can be used to achieve further compression using variable length coding (VLC). Frequently occurring values are converted to short code words, and infrequent values are converted to long code words. To aid deserialization, no code word can be the prefix of another.

2.6 A Spatial Coder

Figure 2-7 ties together all of the preceding spatial coding concepts. The input signal is assumed to be 4:2:2 SDI (Serial Digital Interface), which may have 8- or 10-bit wordlength. MPEG uses only 8-bit resolution; therefore, a rounding stage will be needed when the SDI signal contains 10-bit words. Most MPEG profiles operate with 4:2:0 sampling; therefore, a vertical low-pass filter/interpolation stage will be needed. Rounding and color subsampling introduces a small irreversible loss of information and a proportional reduction in bit rate. The raster scanned input format will need to be stored so that it can be converted to 8 x 8 pixel blocks.



► Figure 2-7.



► Figure 2-8.

The DCT stage transforms the picture information to the frequency domain. The DCT itself does not achieve any compression. Following DCT, the coefficients are weighted and truncated, providing the first significant compression. The coefficients are then zig-zag scanned to increase the probability that the significant coefficients occur early in the scan. After the last non-zero coefficient, an EOB (end of block) code is generated.

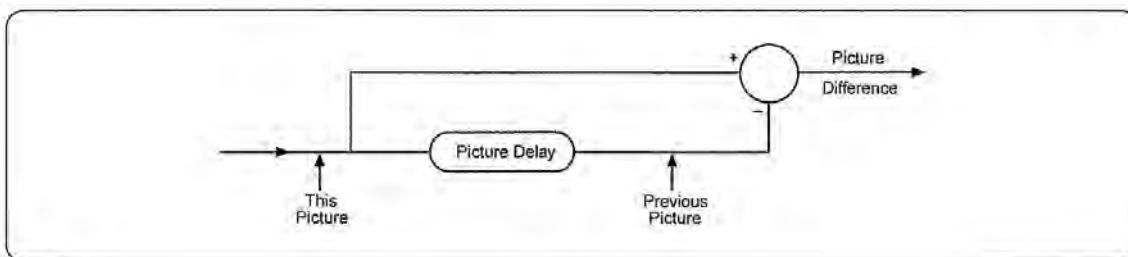
Coefficient data are further compressed by run-length and variable-length coding. In a variable bit-rate system, the quantizing is fixed, but in a fixed bit-rate system, a buffer memory is used to absorb variations in coding difficulty. Highly detailed pictures will tend to fill the buffer, whereas plain pictures will allow it to empty. If the buffer is in danger of overflowing, the requantizing steps will have to be made larger, so that the compression factor is effectively raised.

In the decoder, the bit stream is deserialized and the entropy coding is reversed to reproduce the weighted coefficients. The inverse weighting is applied and coefficients are placed in the matrix according to the zig-zag scan to recreate the DCT matrix. Following an inverse transform, the 8 x 8 pixel block is recreated. To obtain a raster-scanned output, the blocks are stored in RAM, which is read a line at a time. To obtain a 4:2:2 output from 4:2:0 data, a vertical interpolation process will be needed as shown in Figure 2-8.

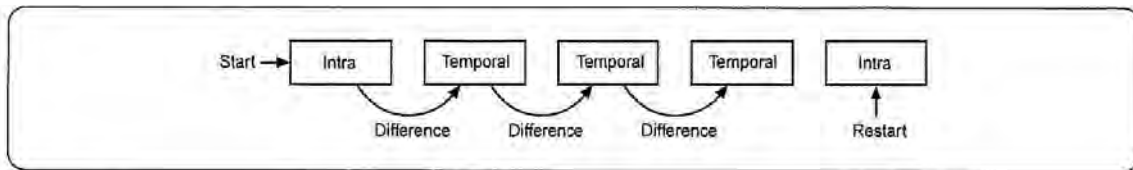
The chroma samples in 4:2:0 are positioned half way between luminance samples in the vertical axis so that they are evenly spaced when an interlaced source is used.

2.7 Temporal Coding

Temporal redundancy can be exploited by inter-coding or transmitting only the differences between pictures. Figure 2-9 shows that a one-picture delay com-



► Figure 2-9.



► Figure 2-10.

bined with a subtractor can compute the picture differences. The picture difference is an image in its own right and can be further compressed by the spatial coder as was previously described. The decoder reverses the spatial coding and adds the difference picture to the previous picture to obtain the next picture.

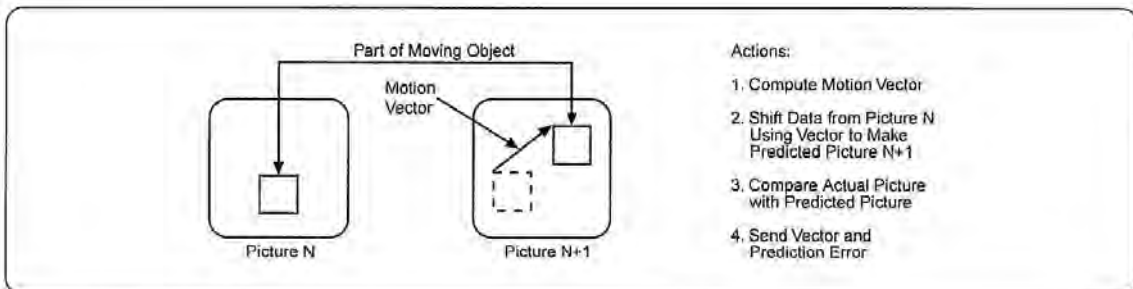
There are some disadvantages to this simple system. First, as only differences are sent, it is impossible to begin decoding after the start of the transmission. This limitation makes it difficult for a decoder to provide pictures following a switch from one bit stream to another (as occurs when the viewer changes channels). Second, if any part of the difference data is incorrect, the error in the picture will propagate indefinitely.

The solution to these problems is to use a system that is not completely differential. Figure 2-10 shows that periodically complete pictures are sent. These

are called Intra-coded pictures (or I-pictures), and they are obtained by spatial compression only. If an error or a channel switch occurs, it will be possible to resume correct decoding at the next I-picture.

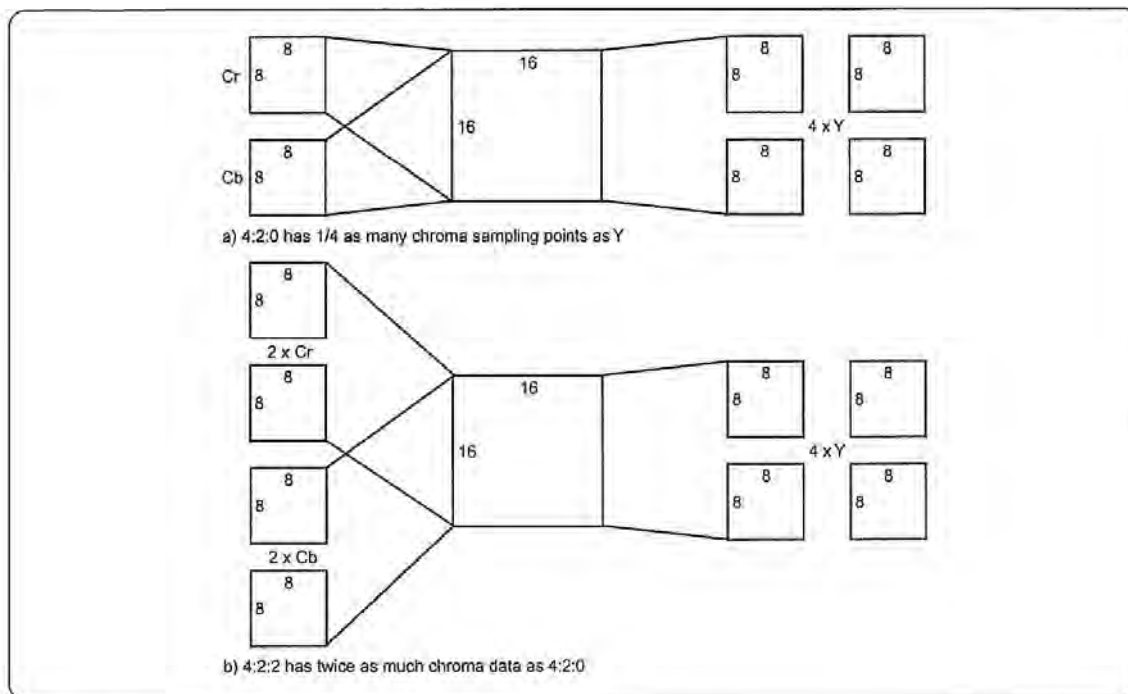
2.8 Motion Compensation

Motion reduces the similarities between pictures and increases the data needed to create the difference picture. Motion compensation is used to increase the similarity. Figure 2-11 shows the principle. When an object moves across the TV screen, it may appear in a different place in each picture, but it does not change in appearance very much. The picture difference can be reduced by measuring the motion at the encoder. This is sent to the decoder as a vector. The decoder uses the vector to shift part of the previous picture to a more appropriate place in the new picture.



► Figure 2-11.

A Guide to MPEG Fundamentals and Protocol Analysis



► Figure 2-12.

One vector controls the shifting of an entire area of the picture that is known as a macroblock. The size of the macroblock is determined by the DCT coding and the color subsampling structure. Figure 2-12a shows that, with a 4:2:0 system, the vertical and horizontal spacing of color samples is exactly twice the spacing of luminance. A single 8 x 8 DCT block of color samples extends over the same area as four 8 x 8 luminance blocks; therefore this is the minimum picture area which can be shifted by a vector. One 4:2:0 macroblock contains four luminance blocks: one Cr block and one Cb block.

In the 4:2:2 profile, color is only subsampled in the horizontal axis. Figure 2-12b shows that in 4:2:2, a single 8 x 8 DCT block of color samples extends over two luminance blocks. A 4:2:2 macroblock contains four luminance blocks: two Cr blocks and two Cb blocks.

The motion estimator works by comparing the luminance data from two successive pictures. A macroblock in the first picture is used as a reference. When the input is interlaced, pixels will be at different vertical locations in the two

fields, and it will, therefore, be necessary to interpolate one field before it can be compared with the other. The correlation between the reference and the next picture is measured at all possible displacements with a resolution of half a pixel over the entire search range. When the greatest correlation is found, this correlation is assumed to represent the correct motion.

The motion vector has a vertical and horizontal component. In typical program material, a moving object may extend over a number of macroblocks. A greater compression factor is obtained if the vectors are transmitted differentially. When a large object moves, adjacent macroblocks have the same vectors and the vector difference becomes zero.

Motion vectors are associated with macroblocks, not with real objects in the image and there will be occasions where part of the macroblock moves and part of it does not. In this case, it is impossible to compensate properly. If the motion of the moving part is compensated by transmitting a vector, the stationary part will be incorrectly shifted, and it will need difference data to be cor-

(14) www.tektronix.com/video_audio/

rected. If no vector is sent, the stationary part will be correct, but difference data will be needed to correct the moving part. A practical compressor might attempt both strategies and select the one which required the least difference data.

2.9 Bidirectional Coding

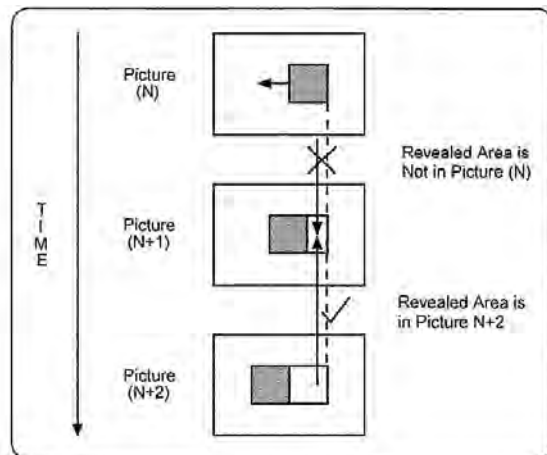
When an object moves, it conceals the background at its leading edge and reveals the background at its trailing edge. The revealed background requires new data to be transmitted because the area of background was previously concealed and no information can be obtained from a previous picture. A similar problem occurs if the camera pans: new areas come into view and nothing is known about them. MPEG helps to minimize this problem by using bidirectional coding, which allows information to be taken from pictures before and after the current picture. If a background is being revealed, it will be present in a later picture, and the information can be moved backwards in time to create part of an earlier picture.

Figure 2-13 shows the concept of bidirectional coding. On an individual macro-block basis, a bidirectionally coded picture can obtain motion-compensated data from an earlier or later picture, or even use an average of earlier and later data. Bidirectional coding significantly reduces the amount of difference data needed by improving the degree of prediction possible. MPEG does not specify how an encoder should be built, only what constitutes a compliant bit stream. However, an intelligent compressor could try all three coding strategies and select the one that results in the least data to be transmitted.

2.10 I, P, and B Pictures

In MPEG, three different types of pictures are needed to support differential and bidirectional coding while minimizing error propagation:

I pictures are Intra-coded pictures that need no additional information for decoding. They require a lot of data compared to other picture types, and therefore they are not transmitted any more frequently than necessary. They

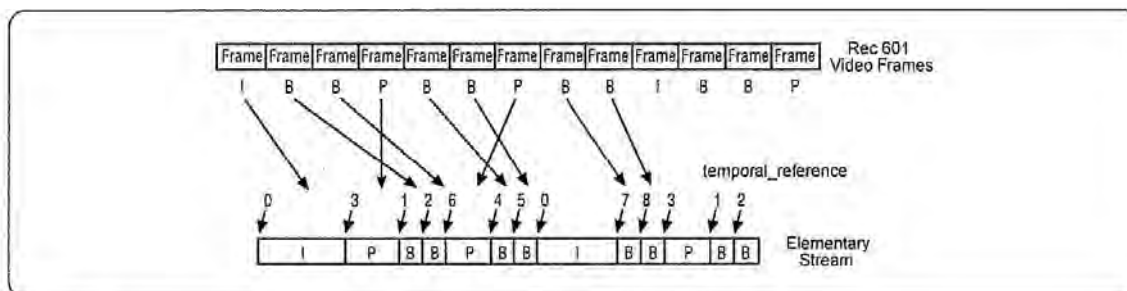


► Figure 2-13.

consist primarily of transform coefficients and have no vectors. I pictures allow the viewer to switch channels, and they arrest error propagation.

P pictures are forward Predicted from an earlier picture, which could be an I picture or a P picture. P-picture data consists of vectors describing where, in the previous picture, each macro-block should be taken from, and transform coefficients that describe the correction or difference data that must be added to that macroblock. P pictures require roughly half the data of an I picture.

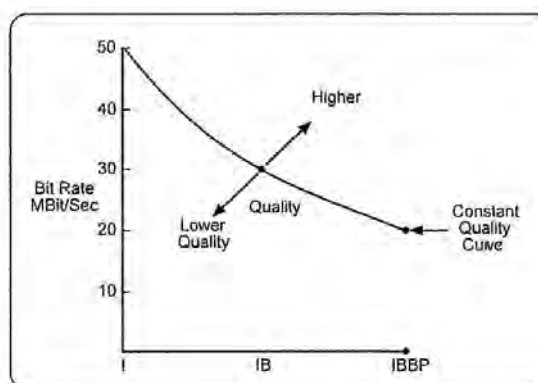
B pictures are Bidirectionally predicted from earlier or later I or P pictures. B-picture data consists of vectors describing where in earlier or later pictures data should be taken from. It also contains the transform coefficients that provide the correction. Because bidirectional prediction is so effective, the correction data are minimal and this helps the B picture to typically require one quarter the data of an I picture.



► Figure 2-14.

Figure 2-14 introduces the concept of the GOP or Group Of Pictures. The GOP begins with an I picture and then has P pictures spaced throughout. The remaining pictures are B pictures. The GOP is defined as ending at the last picture before the next I picture. The GOP length is flexible, but 12 or 15 pictures is a common value. Clearly, if data for B pictures are to be taken from a future picture, that data must already be available at the decoder. Consequently, bidirectional coding requires that picture data is sent out of sequence and temporarily stored. Figure 2-14 also shows that the P-picture data are sent before the B-picture data. Note that the last B pictures in the GOP cannot be transmitted until after the I picture of the next GOP since this data will be needed to bidirectionally decode them. In order to return pictures to their correct sequence, a temporal reference is included with each picture. As the picture rate is also embedded periodically in headers in the bit stream, an MPEG file may be displayed by a personal computer, for example, in the correct order and timescale.

Sending picture data out of sequence requires additional memory at the encoder and decoder and also causes delay. The number of bidirectionally coded pictures between intra- or forward-predicted pictures must be restricted to reduce cost and minimize delay, if delay is an issue.



► Figure 2-15.

Figure 2-15 shows the tradeoff that must be made between compression factor and coding delay. For a given quality, sending only I pictures requires more than twice the bit rate of an IBBP sequence. Where the ability to edit is important, an IB sequence is a useful compromise.

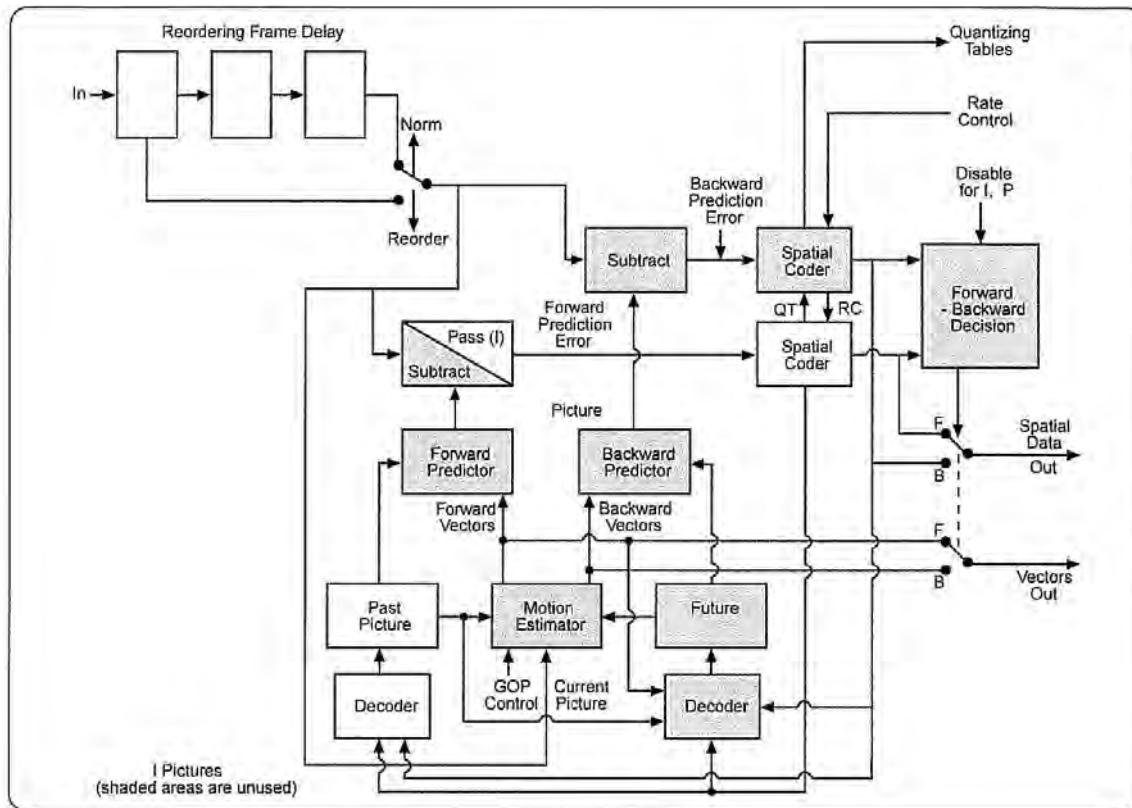
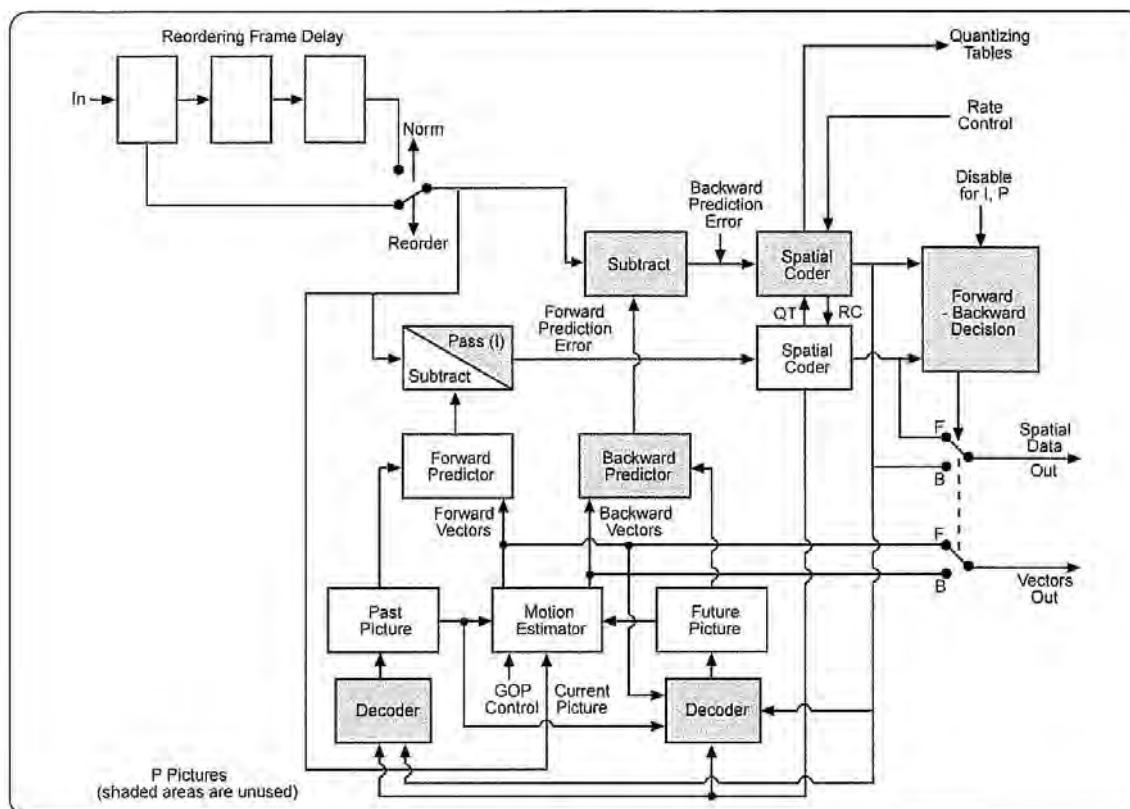


Figure 2-16a.

2.11 An MPEG Compressor

Figures 2-16a, b, and c show a typical bidirectional motion compensator structure. Pre-processed input video enters a series of frame stores that can be bypassed to change the picture order. The data then enter the subtractor and

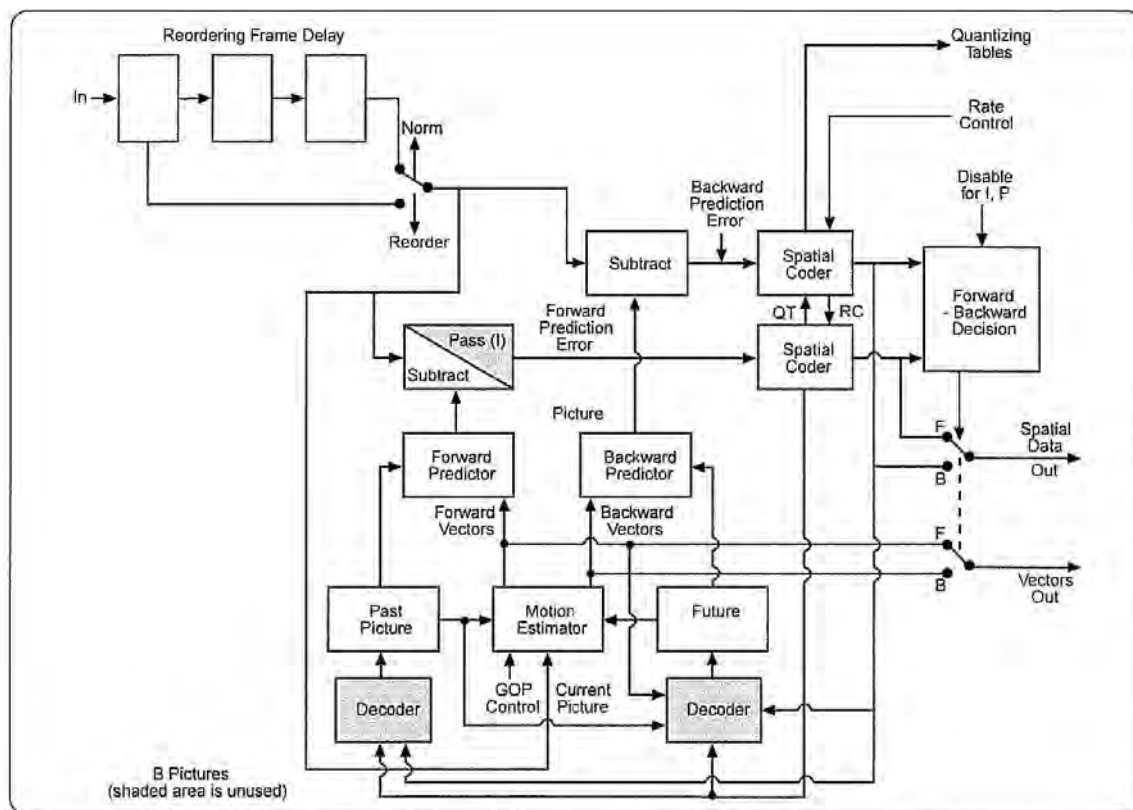
the motion estimator. To create an I picture, the end of the input delay is selected and the subtractor is turned off so that the data pass straight through to be spatially coded (see Figure 2-16a). Subtractor output data also pass to a frame store that can hold several pictures. The I picture is held in the store.



► Figure 2-16b.

To encode a P picture, the B pictures in the input buffer are bypassed, so that the future picture is selected (see Figure 2-16b). The motion estimator compares the I picture in the output store with the P picture in the input store to create forward motion vectors. The I picture is shifted by these vectors to make

a predicted P picture. The predicted P picture is subtracted from the actual P picture to produce the prediction error, which is spatially coded and sent along with the vectors. The prediction error is also added to the predicted P picture to create a locally decoded P picture that also enters the output store.

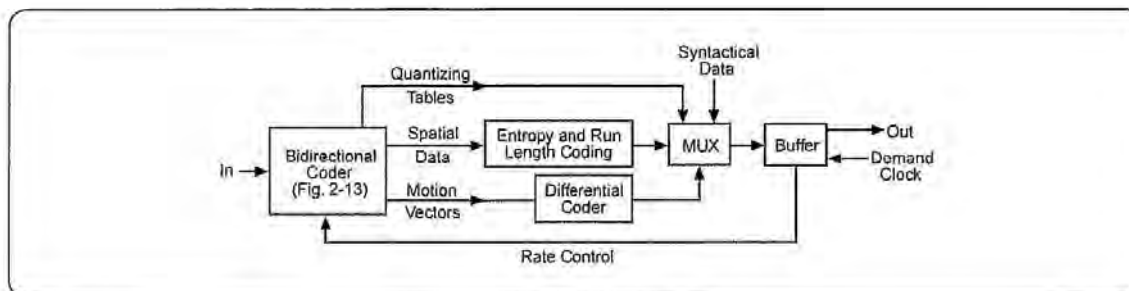


► Figure 2-16c.

The output store then contains an I picture and a P picture. A B picture from the input buffer can now be selected. The motion compensator will compare the B picture with the I picture that precedes it and the P picture that follows it to obtain bidirectional vectors (see Figure 2-16c). Forward and backward motion compensation is performed to produce two predicted B pictures. These are subtracted from the current B picture. On a macroblock-by-macroblock

basis, the forward or backward data are selected according to which represent the smallest differences. The picture differences are then spatially coded and sent with the vectors.

When all of the intermediate B pictures are coded, the input memory will once more be bypassed to create a new P picture based on the previous P picture.



► Figure 2-17.

Figure 2-17 shows an MPEG coder. The motion compensator output is spatially coded and the vectors are added in a multiplexer. Syntactical data is also added which identifies the type of picture (I, P, or B) and provides other information to help a decoder (see **Section 4 – Elementary Streams**). The output data are buffered to allow temporary variations in bit rate. If the bit rate shows a long term increase, the buffer will tend to fill up; to prevent overflow, the quantization process will have to be made more severe. Equally, should the buffer show signs of underflow, the quantization will be relaxed to maintain the average bit rate. This means that the store contains exactly what the store in the decoder will contain, so that the results of all previous coding errors are present. These will automatically be reduced when the predicted picture is subtracted from the actual picture because the difference data will be more accurate.

2.12 Preprocessing

A compressor attempts to eliminate redundancy within the picture and between pictures. Anything which reduces that redundancy is undesirable. Noise and film grain are particularly problematic because they generally occur over the entire picture. After the DCT process, noise results in more non-zero coefficients, which the coder cannot distinguish from genuine picture data. Heavier quantizing will be required to encode all of the coefficients, reducing picture quality. Noise also reduces similarities between successive pictures, increasing the difference data needed.

Residual subcarrier in video decoded from composite video is a serious problem because it results in high, spatial frequencies that are normally at a low level in component programs. Subcarrier also alternates from picture to picture causing an increase in difference data. Naturally, any composite decoding artifact that is visible in the input to the MPEG coder is likely to be reproduced at the decoder.

Any practice that causes unwanted motion is to be avoided. Unstable camera mountings, in addition to giving a shaky picture, increase picture differences and vector transmission requirements. This will also happen with telecine material if film weave or hop due to sprocket hole damage is present. In general, video that is to be compressed must be of the highest quality possible. If high quality cannot be achieved, then noise reduction and other stabilization techniques will be desirable.

If a high compression factor is required, the level of artifacts can increase, especially if input quality is poor. In this case, it may be better to reduce the entropy entering the coder using prefiltering. The video signal is subject to two-dimensional, low-pass filtering, which reduces the number of coefficients needed and reduces the level of artifacts. The picture will be less sharp, but less sharpness is preferable to a high level of artifacts.

In most MPEG-2 applications, 4:2:0 sampling is used, which requires a chroma downsampling process if the source is 4:2:2. In MPEG-1, the luminance and chroma are further downsampled to produce an input picture or SIF (Source Input Format), that is only 352-pixels wide. This technique reduces the entropy by a further factor. For very high compression, the QSIF (Quarter Source Input Format) picture, which is 176-pixels wide, is used. Downsampling is a process that combines a spatial low-pass filter with an interpolator. Downsampling interlaced signals is problematic because vertical detail is spread over two fields which may de-correlate due to motion.

When the source material is telecine, the video signal has different characteristics than normal video. In 50 Hz video, pairs of fields represent the same film frame, and there is no motion between them. Thus, the motion between fields alternates between zero and the motion between frames. This causes vector data to alternate between zero and twice what it should be, varying the bit rate available for picture data. In 60 Hz video, 3:2 pulldown is used to obtain 60 Hz

HIGH		4:2:0 1920x1152 80 Mb/s I,P,B				4:2:0, 4:2:2 1920x1152 100 Mb/s I,P,B
HIGH-1440		4:2:0 1440x1152 60 Mb/s I,P,B			4:2:0 1440x1152 60 Mb/s I,P,B	4:2:0, 4:2:2 1440x1152 80 Mb/s I,P,B
MAIN	4:2:0 720x576 15 Mb/s I,P	4:2:0 720x576 15 Mb/s I,P,B	4:2:2 720x608 50 Mb/s I,P,B	4:2:0 720x576 15 Mb/s I,P,B		4:2:0, 4:2:2 720x576 20 Mb/s I,P,B
LOW		4:2:0 352x288 4 Mb/s I,P,B		4:2:0 352x288 4 Mb/s I,P,B		
LEVEL PROFILE	SIMPLE	MAIN	4:2:2 PROFILE	SNR	SPATIAL	HIGH

Figure 2-18.

from 24 Hz film. One frame is made into two fields, the next is made into three fields, and so on.

Consequently, one field in five is completely redundant, MPEG handles film material best by discarding the third field in 3:2 systems. A 24 Hz code in the transmission alerts the decoder to recreate the 3:2 sequence by re-reading a field store. In 50 and 60 Hz telecine, pairs of fields are deinterlaced to create frames, and then motion is measured between frames. The decoder can recreate interlace by reading alternate lines in the frame store.

A cut is a difficult event for a compressor to handle because it results in an almost complete prediction failure, requiring a large amount of correction data. If a coding delay can be tolerated, a coder may detect cuts in advance and modify the GOP structure dynamically, so that the cut is made to coincide with the generation of an I picture. In this case, the cut is handled with very little extra data. The last 8 pictures before the I frame will almost certainly need to use forward prediction. In some applications that are not real-time, such as DVD mastering, a coder could take two passes at the input video: one pass to identify the difficult or high entropy areas and create a coding strategy, and a second pass to actually compress the input video.

2.13 Profiles and Levels

MPEG is applicable to a wide range of applications requiring different performance and complexity. Using all of the encoding tools defined in MPEG, there are millions of combinations possible. For practical purposes, the MPEG-2 standard is divided into profiles, and each profile is subdivided into levels (see Figure

2-18). A profile is basically a subset of the entire coding repertoire requiring a certain complexity. A level is a parameter such as the size of the picture or bit rate used with that profile. In principle, there are 24 combinations, but not all of these have been defined. An MPEG decoder having a given Profile and Level must also be able to decode lower profiles and levels.

The simple profile does not support bidirectional coding, and so only I and P pictures will be output. This reduces the coding and decoding delay and allows simpler hardware. The simple profile has only been defined at Main level (SP@ML).

The Main Profile is designed for a large proportion of uses. The low level uses a low resolution input having only 352 pixels per line. The majority of broadcast applications will require the MP@ML (Main Profile at Main Level) subset of MPEG, which supports SDTV (Standard Definition TV).

The high-1440 level is a high definition scheme that doubles the definition compared to the main level. The high level not only doubles the resolution but maintains that resolution with 16:9 format by increasing the number of horizontal samples from 1440 to 1920.

In compression systems using spatial transforms and requantizing, it is possible to produce scaleable signals. A scaleable process is one in which the input results in a main signal and a "helper" signal. The main signal can be decoded alone to give a picture of a certain quality, but if the information from the helper signal is added, some aspect of the quality can be improved.

A Guide to MPEG Fundamentals and Protocol Analysis

For example, a conventional MPEG coder, by heavily quantizing coefficients, encodes a picture with moderate signal-to-noise ratio results. If, however, that picture is locally decoded and subtracted pixel-by-pixel from the original, a quantizing noise picture results. This picture can be compressed and transmitted as the helper signal. A simple decoder only decodes the main, noisy bit stream, but a more complex decoder can decode both bit streams and combine them to produce a low-noise picture. This is the principle of SNR scalability.

As an alternative, coding only the lower spatial frequencies in a HDTV picture can produce a main bit stream that an SDTV receiver can decode. If the lower definition picture is locally decoded and subtracted from the original picture, a definition-enhancing picture would result. This picture can be coded into a helper signal. A suitable decoder could combine the main and helper signals to recreate the HDTV picture. This is the principle of Spatial scalability.

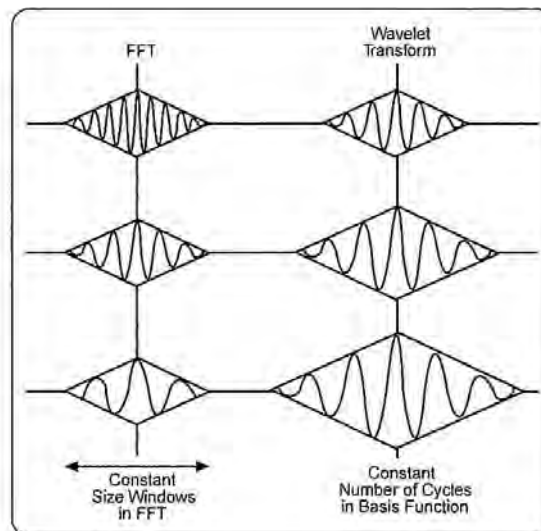
The High profile supports both SNR and spatial scalability as well as allowing the option of 4:2:2 sampling.

The 4:2:2 profile has been developed for improved compatibility with digital production equipment. This profile allows 4:2:2 operation without requiring the additional complexity of using the high profile. For example, an HP@ML decoder must support SNR scalability, which is not a requirement for production. The 4:2:2 profile has the same freedom of GOP structure as other profiles, but in practice it is commonly used with short GOPs making editing easier. 4:2:2 operation requires a higher bit rate than 4:2:0, and the use of short GOPs requires an even higher bit rate for a given quality.

2.14 Wavelets

All transforms suffer from uncertainty because the more accurately the frequency domain is known, the less accurately the time domain is known (and vice versa). In most transforms such as DFT and DCT, the block length is fixed, so the time and frequency resolution is fixed. The frequency coefficients represent evenly spaced values on a linear scale. Unfortunately, because human senses are logarithmic, the even scale of the DFT and DCT gives inadequate frequency resolution at one end and excess resolution at the other.

The wavelet transform is not affected by this problem because its frequency resolution is a fixed fraction of an octave and therefore has a logarithmic char-



► Figure 2-19.

acteristic. This is done by changing the block length as a function of frequency. As frequency goes down, the block becomes longer. Thus, a characteristic of the wavelet transform is that the basis functions all contain the same number of cycles, and these cycles are simply scaled along the time axis to search for different frequencies. Figure 2-19 contrasts the fixed block size of the DFT/DCT with the variable size of the wavelet.

Wavelets are especially useful for audio coding because they automatically adapt to the conflicting requirements of the accurate location of transients in time and the accurate assessment of pitch in steady tones.

For video coding, wavelets have the advantage of producing resolution scaleable signals with almost no extra effort. In moving video, the advantages of wavelets are offset by the difficulty of assigning motion vectors to a variable size block, but in still-picture or I-picture coding this difficulty is not an issue.

► Section 3 - Audio Compression

Lossy audio compression is based entirely on the characteristics of human hearing, which must be considered before any description of compression is possible. Surprisingly, human hearing, particularly in stereo, is actually more critically discriminating than human vision, and consequently audio compression should be undertaken with care. As with video compression, audio compression requires a number of different levels of complexity according to the required compression factor.

3.1 The Hearing Mechanism

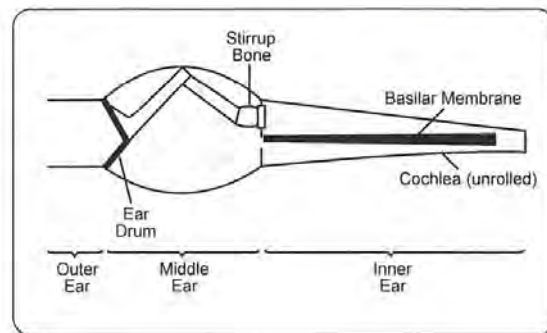
Hearing comprises physical processes in the ear and nervous/mental processes that combine to give us an impression of sound. The impression we receive is not identical to the actual acoustic waveform present in the ear canal because some entropy is lost. Audio compression systems that lose only that part of the entropy that will be lost in the hearing mechanism will produce good results.

The physical hearing mechanism consists of the outer, middle, and inner ears. The outer ear comprises the ear canal and the eardrum. The eardrum converts the incident sound into a vibration in much the same way as does a microphone diaphragm. The inner ear works by sensing vibrations transmitted through a fluid. The impedance of fluid is much higher than that of air and the middle ear acts as an impedance-matching transformer that improves power transfer.

Figure 3-1 shows that vibrations are transferred to the inner ear by the stirrup bone, which acts on the oval window. Vibrations in the fluid in the ear travel up the cochlea, a spiral cavity in the skull (shown unrolled in Figure 3-1 for clarity). The basilar membrane is stretched across the cochlea. This membrane varies in mass and stiffness along its length. At the end near the oval window, the membrane is stiff and light, so its resonant frequency is high. At the distant end, the membrane is heavy and soft and resonates at low frequency. The range of resonant frequencies available determines the frequency range of human hearing, which in most people is from 20 Hz to about 15 kHz.

Different frequencies in the input sound cause different areas of the membrane to vibrate. Each area has different nerve endings to allow pitch discrimination. The basilar membrane also has tiny muscles controlled by the nerves that together act as a kind of positive-feedback system that improves the Q factor of the resonance.

The resonant behavior of the basilar membrane is an exact parallel with the behavior of a transform analyzer. According to the uncertainty theory of trans-



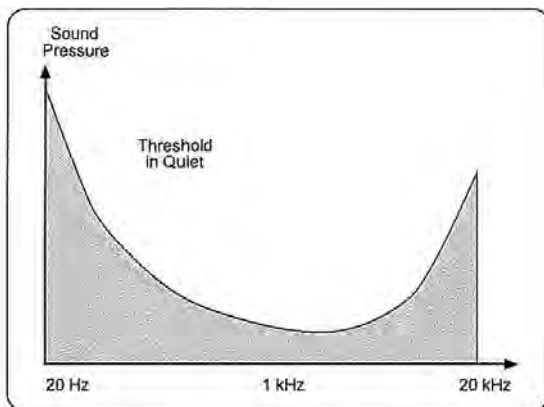
► Figure 3-1.

forms, the more accurately the frequency domain of a signal is known, the less accurately the time domain is known. Consequently, the more able a transform is able to discriminate between two frequencies, the less able it is to discriminate between the time of two events. Human hearing has evolved with a certain compromise that balances time-uncertainty discrimination and frequency discrimination; in the balance, neither ability is perfect.

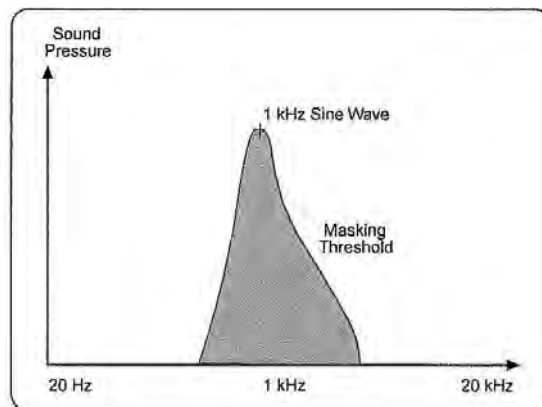
The imperfect frequency discrimination results in the inability to separate closely-spaced frequencies. This inability is known as auditory masking, defined as the reduced sensitivity to sound in the presence of another.

Figure 3-2a shows that the threshold of hearing is a function of frequency. The greatest sensitivity is, not surprisingly, in the speech range. In the presence of a single tone, the threshold is modified as in Figure 3-2b. Note that the threshold is raised for tones at higher frequency and to some extent at lower frequency. In the presence of a complex input spectrum, such as music, the threshold is raised at nearly all frequencies. One consequence of this behavior is that the hiss from an analog audio cassette is only audible during quiet passages in music. Companding makes use of this principle by amplifying low-level audio signals prior to recording or transmission and returning them to their correct level afterwards.

The imperfect time discrimination of the ear is due to its resonant response. The Q factor is such that a given sound has to be present for at least about 1 millisecond before it becomes audible. Because of this slow response, masking can still take place even when the two signals involved are not simultaneous.



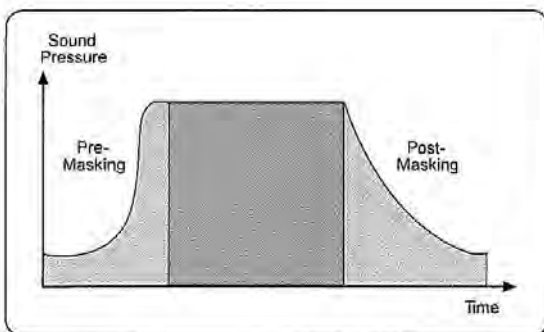
► Figure 3-2a.



► Figure 3-2b.

Forward and backward masking occur when the masking sound continues to mask sounds at lower levels before and after the masking sound's actual duration. Figure 3-3 shows this concept.

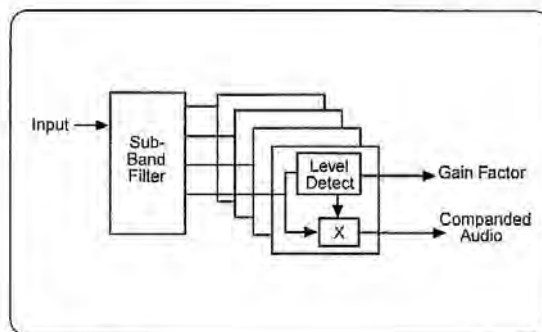
Masking raises the threshold of hearing, and compressors take advantage of this effect by raising the noise floor, which allows the audio waveform to be expressed with fewer bits. The noise floor can only be raised at frequencies at which there is effective masking. To maximize effective masking, it is necessary to split the audio spectrum into different frequency bands to allow introduction of different amounts of companding and noise in each band.



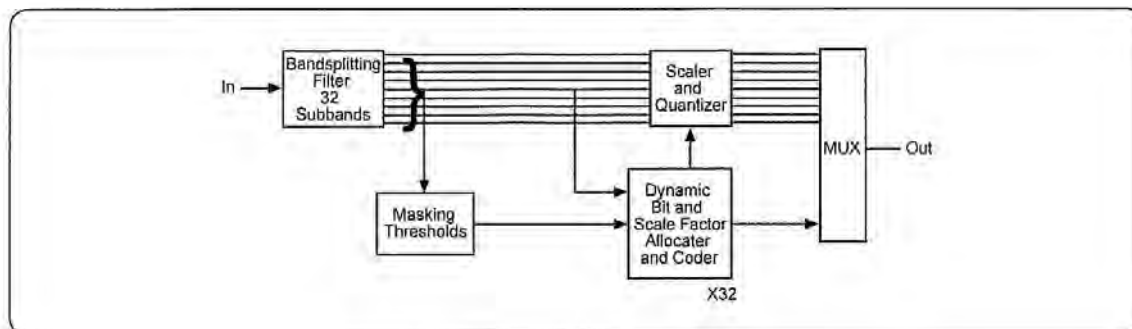
► Figure 3-3.

3.2 Subband Coding

Figure 3-4 shows a band-splitting compandor. The band-splitting filter is a set of narrow-band, linear-phase filters that overlap and all have the same bandwidth. The output in each band consists of samples representing a waveform. In each frequency band, the audio input is amplified up to maximum level prior to transmission. Afterwards, each level is returned to its correct value. Noise picked up in the transmission is reduced in each band. If the noise reduction is compared with the threshold of hearing, it can be seen that greater noise can be tolerated in some bands because of masking. Consequently, in each band after companding, it is possible to reduce the wordlength of samples. This technique achieves a compression because the noise introduced by the loss of resolution is masked.



► Figure 3-4.



► Figure 3-5.

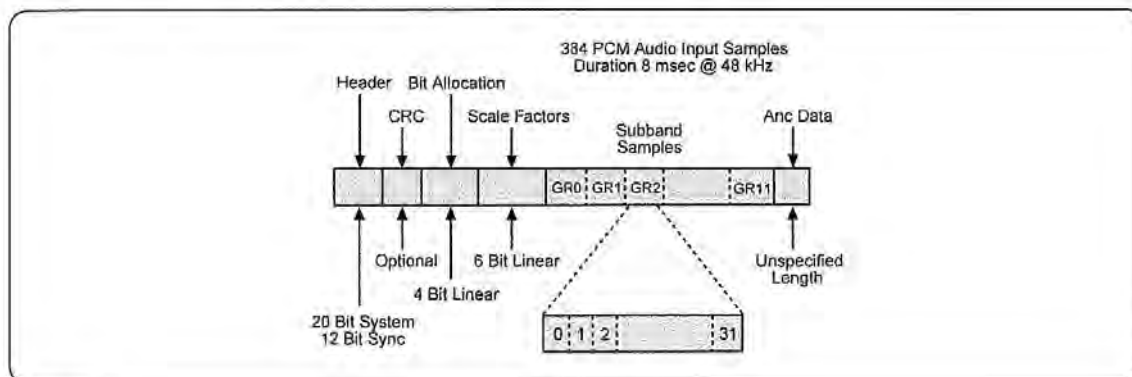
Figure 3-5 shows a simple band-splitting coder as is used in MPEG Layer 1. The digital audio input is fed to a band-splitting filter that divides the spectrum of the signal into a number of bands. In MPEG, this number is 32. The time axis is divided into blocks of equal length. In MPEG layer 1, this is 384 input samples, so there are 12 samples in each of 32 bands in the output of the filter. Within each band, the level is amplified by multiplication to bring the level up to maximum. The gain required is constant for the duration of a block, and a single scale factor is transmitted with each block for each band in order to allow the process to be reversed at the decoder.

The filter bank output is also analyzed to determine the spectrum of the input signal. This analysis drives a masking model that determines the degree of masking that can be expected in each band. The more masking available, the less accurate the samples in each band can be. The sample accuracy is

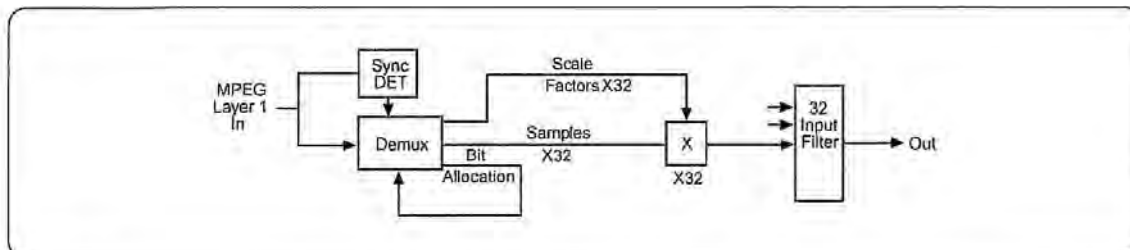
reduced by requantizing to reduce wordlength. This reduction is also constant for every word in a band, but different bands can use different wordlengths. The wordlength needs to be transmitted as a bit allocation code for each band to allow the decoder to deserialize the bit stream properly.

3.3 MPEG Layer 1

Figure 3-6 shows an MPEG Level 1 audio bit stream. Following the synchronizing pattern and the header, there are 32-bit allocation codes of four bits each. These codes describe the wordlength of samples in each subband. Next come the 32 scale factors used in the companding of each band. These scale factors determine the gain needed in the decoder to return the audio to the correct level. The scale factors are followed, in turn, by the audio data in each band.



► Figure 3-6.



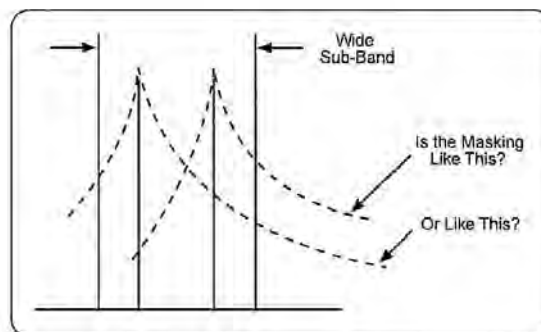
► Figure 3-7.

Figure 3-7 shows the Layer 1 decoder. The synchronization pattern is detected by the timing generator, which deserializes the bit allocation and scale factor data. The bit allocation data then allows deserialization of the variable length samples. The requantizing is reversed and the compression is reversed by the scale factor data to put each band back to the correct level. These 32 separate bands are then combined in a combiner filter which produces the audio output.

3.4 MPEG Layer 2

Figure 3-8 shows that when the band-splitting filter is used to drive the masking model, the spectral analysis is not very accurate, since there are only 32 bands and the energy could be anywhere in the band. The noise floor cannot be raised very much because, in the worst case shown, the masking may not operate. A more accurate spectral analysis would allow a higher compression factor. In MPEG layer 2, the spectral analysis is performed by a separate process. A 512-point FFT working directly from the input is used to drive the masking model instead. To resolve frequencies more accurately, the time span of the transform has to be increased, which is done by raising the block size to 1152 samples.

While the block-companding scheme is the same as in Layer 1, not all of the scale factors are transmitted, since they contain a degree of redundancy on real program material. The scale factor of successive blocks in the same band differs by 2 dB less than 10% of the time, and advantage is taken of this characteristic by analyzing sets of three successive scale factors. On stationary programs, only one scale factor out of three is sent. As transient content increases in a given subband, two or three scale factors will be sent. A scale factor select code is also sent to allow the decoder to determine what has been sent in each subband. This technique effectively halves the scale factor bit rate.



► Figure 3-8.

3.5 Transform Coding

Layers 1 and 2 are based on band-splitting filters in which the signal is still represented as a waveform. However, Layer 3 adopts transform coding similar to that used in video coding. As was mentioned above, the ear performs a kind of frequency transform on the incident sound and because of the Q factor of the basilar membrane, the response cannot increase or reduce rapidly. Consequently, if an audio waveform is transformed into the frequency domain, the coefficients do not need to be sent very often. This principle is the basis of transform coding. For higher compression factors, the coefficients can be requantized, making them less accurate. This process produces noise which will be placed at frequencies where the masking is the greatest. A by-product of a transform coder is that the input spectrum is accurately known, so a precise masking model can be created.

3.6 MPEG Layer 3

This complex level of coding is really only required when the highest compression factor is needed. It has a degree of commonality with Layer 2. A discrete cosine transform is used having 384 output coefficients per block. This output can be obtained by direct processing of the input samples, but in a multi-level coder, it is possible to use a hybrid transform incorporating the 32-band filtering of Layers 1 and 2 as a basis. If this is done, the 32 subbands from the QMF (Quadrature Mirror Filter) are each further processed by a 12-band MDCT (Modified Discrete Cosine Transform) to obtain 384 output coefficients.

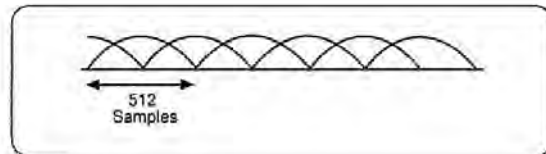
Two window sizes are used to avoid pre-echo on transients. The window switching is performed by the psycho-acoustic model. It has been found that pre-echo is associated with the entropy in the audio rising above the average value. To obtain the highest compression factor, nonuniform quantizing of the coefficients is used along with Huffman coding. This technique allocates the shortest wordlengths to the most common code values.

3.7 AC-3

The AC-3 audio coding technique is used with the ATSC system instead of one of the MPEG audio coding schemes. AC-3 is a transform-based system that obtains coding gain by requantizing frequency coefficients.

The PCM input to an AC-3 coder is divided into overlapping windowed blocks as shown in Figure 3-9. These blocks contain 512 samples each, but because of the complete overlap, there is 100% redundancy. After the transform, there are 512 coefficients in each block, but because of the redundancy, these coefficients can be decimated to 256 coefficients using a technique called Time Domain Aliasing Cancellation (TDAC).

The input waveform is analyzed, and if there is a significant transient in the second half of the block, the waveform will be split into two to prevent pre-echo. In this case, the number of coefficients remains the same, but the fre-



► Figure 3-9.

quency resolution will be halved and the temporal resolution will be doubled. A flag is set in the bit stream to indicate to the decoder that this has been done.

The coefficients are output in floating-point notation as a mantissa and an exponent. The representation is the binary equivalent of scientific notation. Exponents are effectively scale factors. The set of exponents in a block produce a spectral analysis of the input to a finite accuracy on a logarithmic scale called the spectral envelope. This spectral analysis is the input to the masking model that determines the degree to which noise can be raised at each frequency.

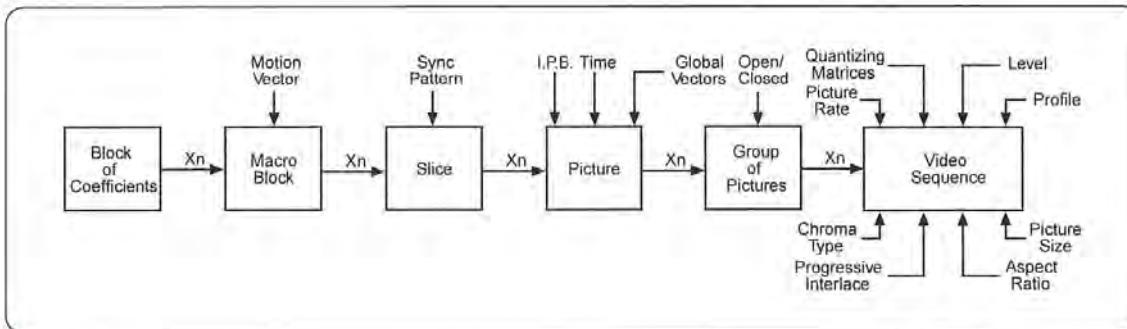
The masking model drives the requantizing process, which reduces the accuracy of each coefficient by rounding the mantissae. A significant proportion of the transmitted data consist of these mantissae.

The exponents are also transmitted, but not directly as there is further redundancy within them that can be exploited. Within a block, only the first (lowest frequency) exponent is transmitted in absolute form. The remainder are transmitted differentially and the decoder adds the difference to the previous exponent. Where the input audio has a smooth spectrum, the exponents in several frequency bands may be the same. Exponents can be grouped into sets of two or four with flags that describe what has been done.

Sets of six blocks are assembled into an AC-3 sync frame. The first block of the frame always has full exponent data, but in cases of stationary signals, later blocks in the frame can use the same exponents.

A Guide to MPEG Fundamentals and Protocol Analysis

► Section 4 - Elementary Streams



► Figure 4-1.

An elementary stream is basically the raw output of an encoder and contains no more than is necessary for a decoder to approximate the original picture or audio. The syntax of the compressed signal is rigidly defined in MPEG so that decoders can be guaranteed to operate on it. The encoder is not defined except that it must somehow produce the right syntax.

The advantage of this approach is that it suits the real world in which there are likely to be many more decoders than encoders. By standardizing the decoder, they can be made at low cost. In contrast, the encoder can be more complex and more expensive without a great cost penalty, but with the potential for better picture quality as complexity increases. When the encoder and the decoder are different in complexity, the coding system is said to be asymmetrical.

The MPEG approach also allows for the possibility that quality will improve as coding algorithms are refined while still producing bit streams that can be understood by earlier decoders. The approach also allows the use of proprietary coding algorithms, which need not enter the public domain.

4.1 Video Elementary Stream Syntax

Figure 4-1 shows the construction of the elementary video stream. The fundamental unit of picture information is the DCT block, which represents an 8 x 8 array of pixels that can be Y, Cr, or Cb. The DC coefficient is sent first and is represented more accurately than the other coefficients. Following the remaining coefficients, an end of block (EOB) code is sent.

Blocks are assembled into macroblocks, which are the fundamental units of a picture and which can be motion compensated. Each macroblock has a two-dimensional motion vector in the header. In B pictures, the vectors can be

backward as well as forward. The motion compensation can be field or frame based and this is indicated. The scale used for coefficient requantizing is also indicated. Using the vectors, the decoder obtains information from earlier and later pictures to produce a predicted picture. The blocks are inverse transformed to produce a correction picture that is added to the predicted picture to produce the decoded output. In 4:2:0 coding, each macroblock will have four Y blocks and two color-difference blocks. To make it possible to identify which block describes which component, the blocks are sent in a specified order.

Macroblocks are assembled into slices that must always represent horizontal strips of picture from left to right. In MPEG, slices can start anywhere and be of arbitrary size, but in ATSC (Advanced Television Systems Committee) they must start at the left-hand edge of the picture. Several slices can exist across the screen width. The slice is the fundamental unit of synchronization for variable length and differential coding. The first vectors in a slice are sent absolutely, whereas the remaining vectors are transmitted differentially. In I pictures, the first DC coefficients in the slice are sent absolutely and the remaining DC coefficients are transmitted differentially. In difference pictures, this technique is not worthwhile.

In the case of a bit error in the elementary stream, either the deserialization of the variable length symbols will break down or subsequent differentially-coded coefficients or vectors will be incorrect. The slice structure allows recovery by providing a resynchronizing point in the bit stream.

A number of slices are combined to make a picture that is the active part of a field or a frame. The picture header defines whether the picture was I, P, or B coded and includes a temporal reference so that the picture can be presented

A Guide to MPEG Fundamentals and Protocol Analysis

at the correct time. In the case of pans and tilts, the vectors in every macro-block will be the same. A global vector can be sent for the whole picture, and the individual vectors then become differences from this global value.

Pictures may be combined to produce a GOP (group of pictures) that must begin with an I picture. The GOP is the fundamental unit of temporal coding. In the MPEG standard, the use of a GOP is optional, but it is a practical necessity. Between I pictures, a variable number of P and/or B pictures may be placed as was described in Section 2. A GOP may be open or closed. In a closed GOP, the last B pictures do not require the I picture in the next GOP for decoding and the bit stream could be cut at the end of the GOP.

If GOPs are used, several GOPs may be combined to produce a video sequence. The sequence begins with a sequence start code, followed by a sequence header and ends with a sequence end code. Additional sequence headers can be placed throughout the sequence. This approach allows decoding to begin part way through the sequence, as might happen in playback of digital video discs and tape cassettes. The sequence header specifies the vertical and horizontal size of the picture, the aspect ratio, the chroma subsampling format, the picture rate, the use of progressive scan or interlace, the profile, level, and bit rate, and the quantizing matrices used in intra and inter-coded pictures.

Without the sequence header data, a decoder cannot understand the bit stream, and therefore sequence headers become entry points at which

decoders can begin correct operation. The spacing of entry points influences the delay in correct decoding that occurs when the viewer switches from one television channel to another.

4.2 Audio Elementary Streams

Various types of audio can be embedded in an MPEG-2 multiplex. These types include audio coded according to MPEG layers 1, 2, 3, or AC-3. The type of audio encoding used must be included in a descriptor that a decoder will read in order to invoke the appropriate type of decoding.

The audio compression process is quite different from the video process. There is no equivalent to the different I, P, and B frame types, and audio frames always contain the same amount of audio data. There is no equivalent of bi-directional coding and audio frames are not transmitted out of sequence.

In MPEG-2 audio, the descriptor in the sequence header contains the layer that has been used to compress the audio and the type of compression used (for example, joint stereo), along with the original sampling rate. The audio sequence is assembled from a number of access units (AUs) which will be coded audio frames.

If AC-3 coding is used, as in ATSC, this usage will be reflected in the sequence header. The audio access unit (AU) is an AC-3 sync frame as described in Section 3.7. The AC-3 sync frame represents a time span equivalent of 1536 audio samples and will be 32 ms for 48 kHz sampling and 48 ms for 32 kHz.

► Section 5 - Packetized Elementary Streams (PES)

For practical purposes, the continuous elementary streams carrying audio or video from compressors need to be broken into packets. These packets are identified by headers that contain time stamps for synchronizing. PES packets can be used to create Program Streams or Transport Streams.

5.1 PES Packets

In the Packetized Elementary Stream (PES), an endless elementary stream is divided into packets of a convenient size for the application. This size might be a few hundred kilobytes, although this would vary with the application.

Each packet is preceded by a PES packet header. Figure 5-1 shows the contents of a header. The packet begins with a start-code prefix of 24 bits and a stream ID that identifies the contents of the packet as video or audio and further specifies the type of audio coding. These two parameters (start code prefix and stream ID) comprise the packet start code that identifies the beginning of a packet. It is important not to confuse the packet in a PES with the much smaller packet used in transport streams that, unfortunately, shares the same name.

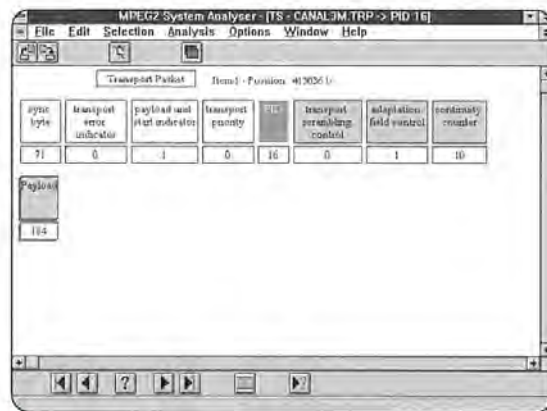
Because MPEG only defines the transport stream, not the encoder, a designer might choose to build a multiplexer that converts from elementary streams to a transport stream in one step. In this case, the PES packets may never exist in an identifiable form, but instead, they are logically present in the transport stream payload.

5.2 Time Stamps

After compression, pictures are sent out of sequence because of bidirectional coding. They require a variable amount of data and are subject to variable delay due to multiplexing and transmission. In order to keep the audio and video locked together, time stamps are periodically incorporated in each picture.

A time stamp is a 33-bit number that is a sample of a counter driven by a 90-kHz clock. This clock is obtained by dividing the 27-MHz program clock by 300. Since presentation times are evenly spaced, it is not essential to include a time stamp in every presentation unit. Instead, time stamps can be interpolated by the decoder, but they must not be more than 700 ms apart in either program streams or transport streams.

Time stamps indicate where a particular access unit belongs in time. Lip sync is obtained by incorporating time stamps into the headers in both video and



► Figure 5-1.

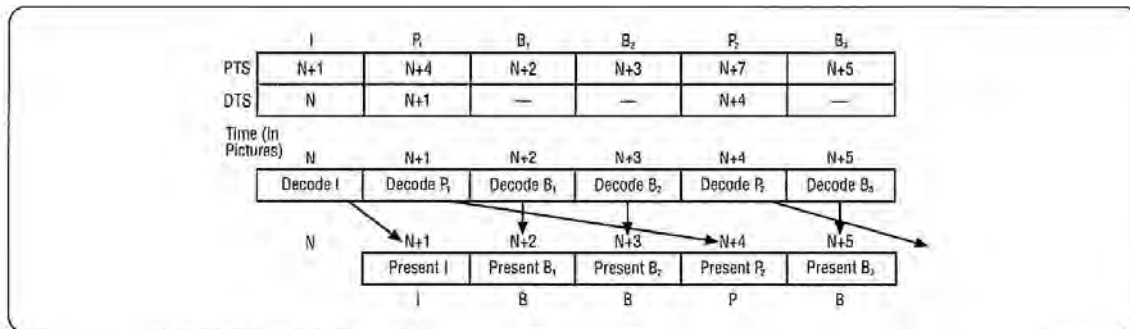
audio PES packets. When a decoder receives a selected PES packet, it decodes each access unit and buffers it into RAM. When the time-line count reaches the value of the time stamp, the RAM is read out. This operation has two desirable results: First, effective timebase correction is obtained in each elementary stream. Second, the video and audio elementary streams can be synchronized together to make a program.

5.3 PTS/DTS

When bidirectional coding is used, a picture may have to be decoded some time before it is presented so that it can act as the source of data for a B picture. Although, for example, pictures can be presented in the order IBPB, they will be transmitted in the order IPBB. Consequently, two types of time stamp exist. The decode time stamp (DTS) indicates the time when a picture must be decoded, whereas a presentation time stamp (PTS) indicates when it must be presented to the decoder output.

B pictures are decoded and presented simultaneously so that they only contain PTS. When an IPBB sequence is received, both I and P must be decoded before the first B picture. A decoder can only decode one picture at a time; therefore the I picture is decoded first and stored. While the P picture is being decoded, the decoded I picture is output so that it can be followed by the B pictures.

A Guide to MPEG Fundamentals and Protocol Analysis



► Figure 5-2.

Figure 5-2 shows that when an access unit containing an I picture is received, it will have both DTS and PTS in the header and these time stamps will be separated by one picture period. If bidirectional coding is being used, a P picture must follow and this picture also has a DTS and a PTS time stamp, but the separation between the two stamp times is three picture periods to allow for the intervening B pictures. Thus, if IPBB is received, I is delayed one picture period, P is delayed three picture periods, the two Bs are not delayed at all,

and the presentation sequence becomes IBBP. Clearly, if the GOP structure is changed such that there are more B pictures between I and P, the difference between DTS and PTS in the P pictures will be greater.

The PTS/DTS flags in the packet header are set to indicate the presence of PTS alone or both PTS and DTS time stamp. Audio packets may contain several access units and the packet header contains a PTS. Because audio packets are never transmitted out of sequence, there is no DTS in an audio packet.

► Section 6 - Program Streams

Program Streams are one way of combining several PES packet streams and are advantageous for recording applications such as DVD.

6.1 Recording vs. Transmission

For a given picture quality, the data rate of compressed video will vary with picture content. A variable bit-rate channel will give the best results. In transmission, most practical channels are fixed and the overall bit-rate is kept constant by the use of stuffing (meaningless data).

In a DVD, the use of stuffing is a waste of storage capacity. However, a storage medium can be slowed down or speeded up, either physically or, in the case of a disk drive, by changing the rate of data transfer requests. This approach allows a variable-rate channel to be obtained without capacity penalty. When a medium is replayed, the speed can be adjusted to keep a data buffer approximately half full, irrespective of the actual bit rate which can change dynamically. If the decoder reads from the buffer at an increased rate, it will tend to empty the buffer, and the drive system will simply increase the access rate to restore balance. This technique only works if the audio and video were encoded from the same clock; otherwise, they will slip over the length of the recording.

To satisfy these conflicting requirements, Program Streams and Transport Streams have been devised as alternatives. A Program Stream works well on a single program with variable bit rate in a recording environment; a Transport Stream works well on multiple programs in a fixed bit-rate transmission environment.

The problem of genlocking to the source does not occur in a DVD player. The player determines the time base of the video with a local SPG (internal or external) and simply obtains data from the disk in order to supply pictures on that time base. In transmission, the decoder has to recreate the time base at the encoder or it will suffer overflow or underflow. Thus, a Transport Stream uses Program Clock Reference (PCR), whereas a Program Stream has no need for the Program Clock.

6.2 Introduction to Program Streams

A Program Stream is a PES packet multiplex that carries several elementary streams that were encoded using the same master clock or system time clock (STC). This stream might be a video stream and its associated audio streams, or a multichannel audio-only program. The elementary video stream is divided into Access Units (AUs), each of which contains compressed data describing one picture. These pictures are identified as I, P, or B and each carries an Access Unit number that indicates the correct display sequence. One video Access Unit becomes one program-stream packet. In video, these packets vary in size. For example, an I picture packet will be much larger than a B picture packet. Digital audio Access Units are generally of the same size and several are assembled into one program-stream packet. These packets should not be confused with transport-stream packets that are smaller and of fixed size. Video and audio Access Unit boundaries rarely coincide on the time axis, but this lack of coincidence is not a problem because each boundary has its own time-stamp structure.

A Guide to MPEG Fundamentals and Protocol Analysis

③4 www.tektronix.com/video_audio/

Google Exhibit No. 1014
Page 040

Google Exhibit No. 1063
Page 0577

► Section 7 - Transport Streams

A transport stream is more than a multiplex of many PES packets. In Program Streams, time stamps are sufficient to recreate the time axis because the audio and video are locked to a common clock. For transmission down a data network over distance, there is an additional requirement to recreate the clock for each program at the decoder. This requires an additional layer of syntax to provide program clock reference (PCR) signals.

7.1 The Job of a Transport Stream

The transport stream carries many different programs and each may use a different compression factor and a bit rate that can change dynamically even though the overall bit rate stays constant. This behavior is called statistical multiplexing and it allows a program that is handling difficult material to borrow bandwidth from a program handling easy material. Each video PES can have a different number of audio and data PESs associated with it. Despite this flexibility, a decoder must be able to change from one program to the next and correctly select the appropriate audio and data channels. Some of the programs can be protected so that they can only be viewed by those who have paid a subscription or fee. The transport stream must contain conditional access (CA) information to administer this protection. The transport stream contains program specific information (PSI) to handle these tasks.

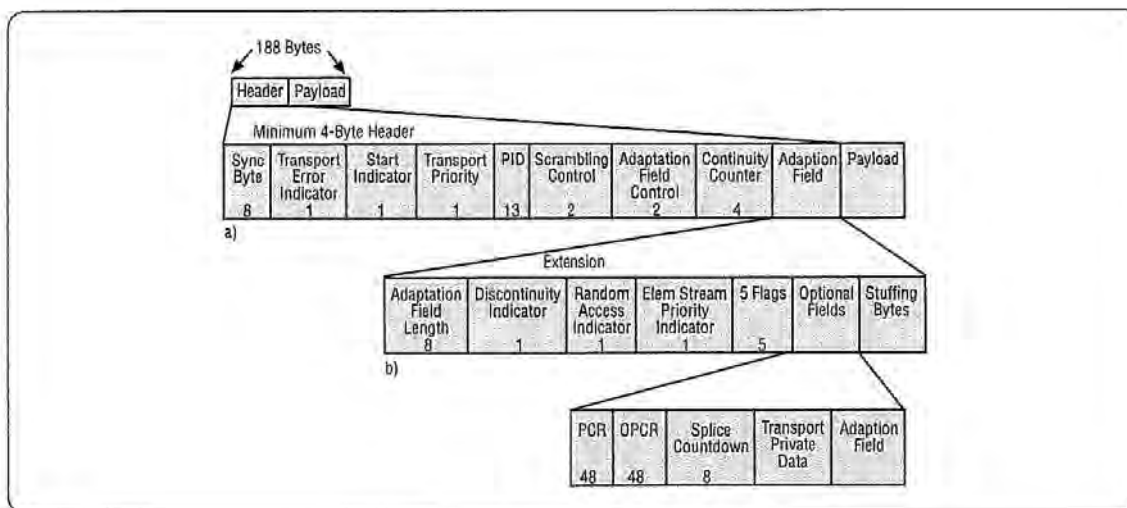
The transport layer converts the PES data into small packets of constant size that are self contained. When these packets arrive at the decoder, there may be

jitter in the timing. The use of time division multiplexing also causes delay, but this factor is not fixed because the proportion of the bit stream allocated to each program is not fixed. Time stamps are part of the solution, but they only work if a stable clock is available. The transport stream must contain further data allowing the recreation of a stable clock.

The operation of digital video production equipment is heavily dependent on the distribution of a stable system clock for synchronization. In video production, genlocking is used, but over long distances, the distribution of a separate clock is not practical. In a transport stream, the different programs may have originated in different places that are not necessarily synchronized. As a result, the transport stream has to provide a separate means of synchronizing for each program.

This additional synchronization method is called a Program Clock Reference (PCR) and it recreates a stable reference clock that can be divided down to create a time line at the decoder, so that the time stamps for the elementary streams in each program stream become useful. Consequently, one definition of a program is a set of elementary streams sharing the same timing reference.

In a Single Program Transport Stream (SPTS), there will be one PCR channel that recreates one program clock for both audio and video. The SPTS is often used as the communication between an audio/video coder and a multiplexer.



► Figure 7-1.

7.2 Packets

Figure 7-1 shows the structure of a transport stream packet. The size is a constant 188 bytes and it is always divided into a header and a payload. Figure 7-1a shows the minimum header of 4 bytes. In this header, the most important information is:

- **The sync byte.** This byte is recognized by the decoder so that the header and the payload can be deserialized.
- **The transport error indicator.** This indicator is set if the error correction layer above the transport layer is experiencing a raw-bit error rate (BER) that is too high to be correctable. It indicates that the packet may contain errors. See **Section 8 – Introduction to DVB/ATSC** for details of the error correction layer.
- **The Packet Identification (PID).** This thirteen-bit code is used to distinguish between different types of packets. More will be said about PID later.
- **The continuity counter.** This four-bit value is incremented by the encoder as each new packet having the same PID is sent. It is used to determine if any packets are lost, repeated, or out of sequence.

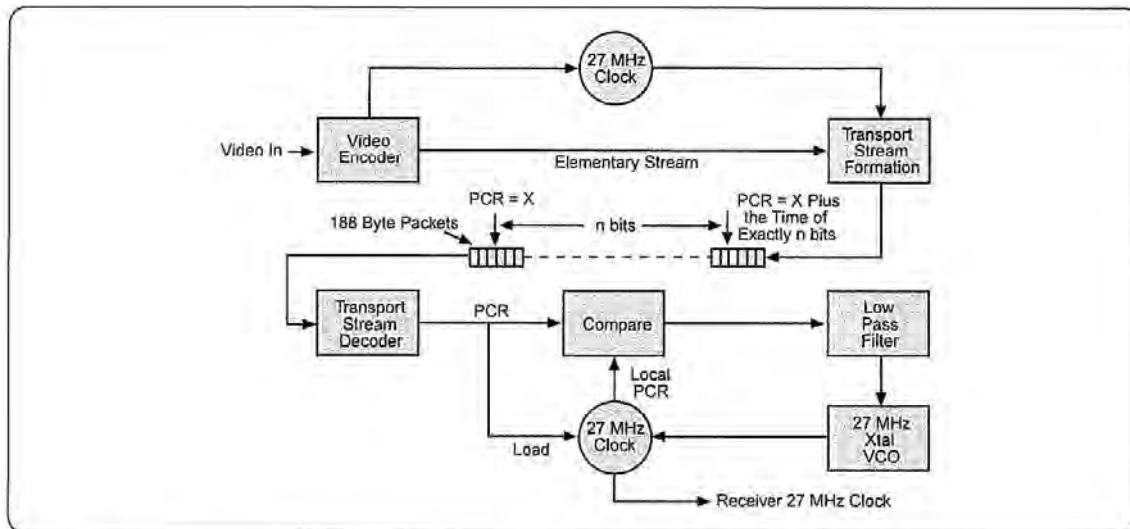
In some cases, more header information is needed, and if this is the case, the adaptation field control bits are set to indicate that the header is larger than normal. Figure 7-1b shows that when this happens the extra header length is described by the adaptation field length code. Where the header is extended, the payload becomes smaller to maintain constant packet length.

7.3 Program Clock Reference (PCR)

The encoder used for a particular program will have a 27 MHz program clock. In the case of an SDI (Serial Digital Interface) input, the bit clock can be divided by 10 to produce the encoder program clock. Where several programs originate in the same production facility, it is possible that they will all have the same clock. In case of an analog video input, the H-sync period will need to be multiplied by a constant in a phase-locked loop to produce 27 MHz.

The adaptation field in the packet header is periodically used to include the PCR (program clock reference) code that allows generation of a locked clock at the decoder. If the encoder or a remultiplexer has to switch sources, the PCR may have a discontinuity. The continuity count can also be disturbed. This event is handled by the discontinuity indicator, which tells the decoder to expect a disturbance. Otherwise, a discontinuity is an error condition.

Figure 7-2 shows how the PCR is used by the decoder to recreate a remote version of the 27-MHz clock for each program. The encoder clocks drive a constantly running binary counter, and the value of these counters are sampled periodically and placed in the header adaptation fields as the PCR. (The PCR, like the PTS, is a 33-bit number that is a sample of a counter driven by a 90-kHz clock). Each encoder produces packets having a different PID. The decoder recognizes the packets with the correct PID for the selected program and ignores others. At the decoder, a VCO generates a nominal 27-MHz clock and this drives a local PCR counter. The local PCR is compared with the PCR



► Figure 7-2.

from the packet header, and the difference is the PCR phase error. This error is filtered to control the VCO that eventually will bring the local PCR count into step with the header PCRs. Heavy VCO filtering ensures that jitter in PCR transmission does not modulate the clock. The discontinuity indicator will reset the local PCR count and, optionally, may be used to reduce the filtering to help the system quickly lock to the new timing.

MPEG requires that PCRs be sent at a rate of at least 10 PCRs per second, whereas DVB specifies a minimum of 25 PCRs per second.

7.4 Packet Identification (PID)

A 13-bit field in the transport packet header contains the Packet Identification Code (PID). The PID is used by the demultiplexer to distinguish between packets containing different types of information. The transport-stream bit rate must be constant, even though the sum of the rates of all of the different streams it contains can vary. This requirement is handled by the use of null packets that contain all zeros in the payload. If the real payload rate falls, more null packets are inserted. Null packets always have the same PID, which is 8191 or thirteen 1s.

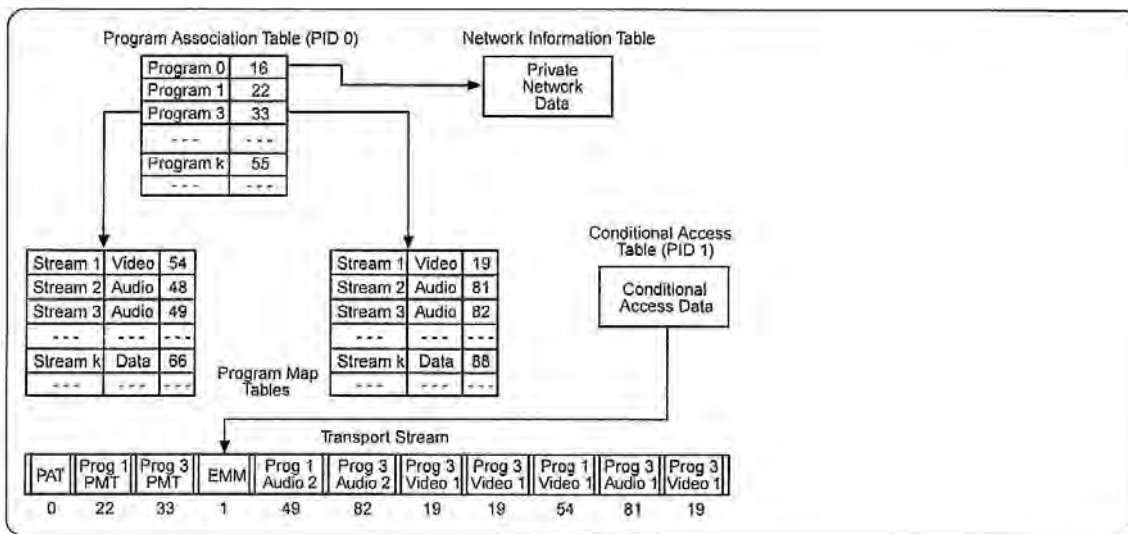
In a given transport stream, all packets belonging to a given elementary stream will have the same PID. Packets in another elementary stream will have another PID. The demultiplexer can easily select all data for a given elementary stream

simply by accepting only packets with the right PID. Data for an entire program can be selected using the PIDs for video, audio, and teletext data. The demultiplexer can correctly select packets only if it can correctly associate them with the transport stream to which they belong. The demultiplexer can do this task only if it knows what the right PIDs are. This is the function of the Program Specific Information (PSI).

7.5 Program Specific Information (PSI)

PSI is carried in packets having unique PIDs, some of which are standardized and some of which are specified by the Program Association Table (PAT) and the Conditional Access Table (CAT). These packets must be included periodically in every transport stream. The PAT always has a PID of 0, and the CAT always has a PID of 1. These values and the null packet PID of 8191 are the only fixed PIDs in the whole MPEG system. The demultiplexer must determine all of the remaining PIDs by accessing the appropriate table. However, in ATSC and DVB, PMTs may require specific PIDs. In this respect (and in some others), MPEG and DVB/ATSC are not fully interchangeable.

The program streams that exist in the transport stream are listed in the program association table (PAT) packets (PID = 0) that specify the PIDs of all Program Map Table (PMT) packets. The first entry in the PAT, program 0, is



► Figure 7-3.

always reserved for network data and contains the PID of network information table (NIT) packets.

The PIDs for Entitlement Control Messages (ECM) and Entitlement Management Messages (EMM) are listed in the Conditional Access Table (CAT) packets (PID = 1).

As Figure 7-3 shows, the PIDs of the video, audio, and data elementary streams that belong in the same program stream are listed in the Program Map Table (PMT) packets. Each PMT packet has its own PID.

A given network information table contains details of more than just the transport stream carrying it. Also included are details of other transport streams that may be available to the same decoder, for example, by tuning to a different RF channel or steering a dish to a different satellite. The NIT may list a number of other transport streams and each one may have a descriptor that specifies the radio frequency, orbital position, and so on. In MPEG, only the NIT is mandatory for this purpose. In DVB, additional metadata, known as DVB-SI, is included, and the NIT is considered to be part of DVB-SI. This operation is discussed in

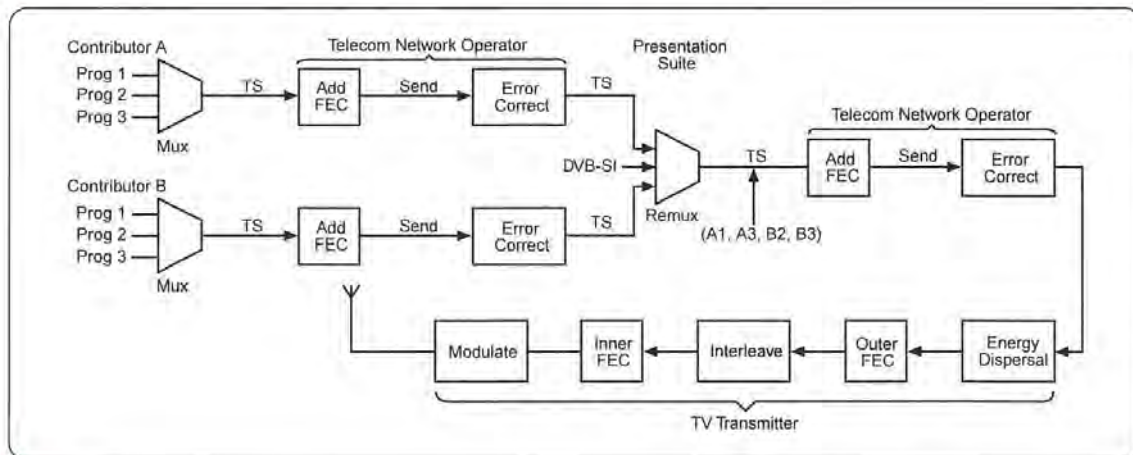
Section 8 – Introduction to DVB/ATSC. When discussing the subject in general, the term PSI/SI is used.

Upon first receiving a transport stream, the demultiplexer must look for PIDs 0 and 1 in the packet headers. All PID 0 packets contain the Program Association Table (PAT). All PID 1 packets contain Conditional Access Table (CAT) data.

By reading the PAT, the demux can find the PIDs of the Network Information Table (NIT) and of each Program Map Table (PMT). By finding the PMTs, the demux can find the PIDs of each elementary stream.

Consequently, if the decoding of a particular program is required, reference to the PAT and then the PMT is all that is needed to find the PIDs of all of the elementary streams in the program. If the program is encrypted, access to the CAT will also be necessary. As demultiplexing is impossible without a PAT, the lockup speed is a function of how often the PAT packets are sent. MPEG specifies a maximum of 0.5 seconds between the PAT packets and the PMT packets that are referred to in those PAT packets. In DVB and ATSC, the NIT may reside in packets that have a specific PID.

► Section 8 - Introduction to DVB/ATSC



► Figure 8-1.

MPEG compression is already being used in broadcasting and will become increasingly important in the future. A discussion of the additional requirements of broadcasting data follows:

8.1 An Overall View

ATSC stands for the Advanced Television Systems Committee, which is a U.S. organization that defines standards for terrestrial digital broadcasting and cable distribution. DVB refers to the Digital Video Broadcasting Project and to the standards and practices established by the DVB Project. This project was originally a European project, but produces standards and guides accepted in many areas of the world. These standards and guides encompass all transmission media, including satellite, cable, and terrestrial broadcasting.

Digital broadcasting has different distribution and transmission requirements, as is shown in Figure 8-1. Broadcasters will produce transport streams that contain several television programs. Transport streams have no protection against errors, and in compressed data, the effect of errors is serious. Transport streams need to be delivered error-free to transmitters, satellite uplinks, and cable head ends. In this context, error free means a BER of 1×10^{-11} . This task is normally entrusted to Telecommunications Network Operators, who will use an additional layer of error correction as needed. This layer should be transparent to the destination.

A particular transmitter or cable operator may not want all of the programs in a transport stream. Several transport streams may be received and a selection of channels may be made and encoded into a single output transport stream using a remultiplexer. The configuration may change dynamically.

Broadcasting in the digital domain consists of conveying the entire transport stream to the viewer. Whether the channel is cable, satellite, or terrestrial, the problems are much the same. Metadata describing the transmission must be encoded into the transport stream in a standardized way. In DVB, this metadata is called Service Information (DVB-SI) and includes details of programs carried on other multiplexes and services such as teletext.

In broadcasting, there is much less control of the signal quality and noise or interference is a possibility. This requires some form of forward error correction (FEC) layer. Unlike the FEC used by the Telecommunications Network Operators, which can be proprietary, (or standardized as per ETSI, which defines DVB transmission over SDH and PDH networks), the FEC used in broadcasting must be standardized so that receivers will be able to handle it.

The addition of error correction obviously increases the bit rate as far as the transmitter or cable is concerned. Unfortunately, reliable, economical radio and cable-transmission of data requires more than serializing the data. Practical systems require channel coding.

8.2 Remultiplexing

Remultiplexing is a complex task because it has to output a compliant bit stream that is assembled from parts of others. The required data from a given input transport stream can be selected with reference to the program association table and the program map tables that will disclose the PIDs of the programs required. It is possible that the same PIDs have been used in two input transport streams; therefore, the PIDs of one or more elementary streams may have to be changed. The packet headers must pass on the program clock reference (PCR) that will allow the final decoder to recreate a 27-MHz clock. As the position of packets containing PCR may be different in the new multiplex, the remultiplexer may need to edit the PCR values to reflect their new position on the time axis.

The program map tables and program association tables will need to be edited to reflect the new transport stream structure, as will the Conditional Access Tables (CAT).

If the sum of the selected program-stream bit rates is less than the output bit rate, the remultiplexer will create stuffing packets with suitable PIDs. However, if the transport streams have come from statistical multiplexers, it is possible that the instantaneous bit rate of the new transport stream will exceed the channel capacity. This condition might occur if several selected programs in different transport streams simultaneously contain high entropy. In this case, the only solution is to recompress and create new, shorter coefficients in one or more bit streams to reduce the bit rate.

8.3 Service Information (SI)

In the future, digital delivery will mean that there will be a large number of programs, teletext, and services available to the viewer and these may be spread across a number of different transport streams. Both the viewer and the Integrated Receiver Decoder (IRD) will need help to display what is available and to output the selected service. This capability requires metadata beyond the capabilities of MPEG-PSI (Program Specific Information) and is referred to as DVB-SI (Service Information). DVB-SI is considered to include the NIT, which must be present in all MPEG transport streams.

DVB-SI is embedded in the transport stream as additional transport packets with unique PIDs and carries technical information for IRDs. DVB-SI also contains Electronic Program Guide (EPG) information, such as the nature of a program, the timing and the channel on which it can be located, and the countries in which it is available. Programs can also be rated so that parental judgment can be exercised.

DVB-SI can include the following options over and above MPEG-PSI:

- ▶ **Service Description Table (SDT).** Each service in a DVB transport stream can have a service descriptor and these descriptors are assembled into the Service Description Table. A service may be television, radio, or teletext. The service descriptor includes the name of the service provider.
- ▶ **Event Information Table (EIT).** EIT is an optional table for DVB which contains program names, start times, durations, and so on.
- ▶ **Bouquet Association Table (BAT).** The BAT is an optional table for DVB that provides details of bouquets, which are collections of services marketed as a single product.
- ▶ **Time and Date Table (TDT).** The TDT is an option that embeds a UTC time and date stamp in the transport stream.

8.4 Error Correction

Error correction is necessary because conditions on long transmission paths cannot be controlled. In some systems, error detection is sufficient because it can be used to request a retransmission. Clearly, this approach will not work with real-time signals such as television. Instead, a system called Forward Error Correction (FEC) is used in which sufficient extra bits, known as redundancy, are added to the data to allow the decoder to perform corrections in real time.

The FEC used in modern systems is usually based on the Reed-Solomon (R-S) codes. A full discussion of these is outside the scope of this book. Briefly, R-S codes add redundancy to the data to make a code-word such that when each symbol is used as a term in a minimum of two simultaneous equations, the sum (or syndrome) is always zero if there is no error. This zero condition is obtained irrespective of the data and makes checking easy. In Transport streams, the packets are always 188 bytes long. The addition of 16 bytes of R-S redundancy produces a standard FEC code-word of 204 bytes.

In the event that the syndrome is non-zero, solving the simultaneous equations will result in two values needed for error correction: the location of the error and the nature of the error. However, if the size of the error exceeds half the amount of redundancy added, the error cannot be corrected. Unfortunately, in typical transmission channels, the signal quality is statistical. This means that while single bits may be in error due to noise, on occasion a large number of bits, known as a burst, can be corrupted together. This corruption might be due to lightning or interference from electrical equipment.

It is not economic to protect every code word against such bursts because they do not occur often enough. The solution is to use a technique known as interleaving. Figure 8-2 shows that, when interleaving is used, the source data are FEC coded, but prior to transmission, they are fed into a RAM buffer. Figure 8-3 shows one possible technique in which data enters the RAM in rows and are then read out in columns. The reordered data are now transmitted. On reception, the data are put back to their original order, or deinterleaved, by using a second RAM. The result of the interleaving process is that a burst of errors in the channel after deinterleaving becomes a large number of single-symbol errors, which are more readily correctable.

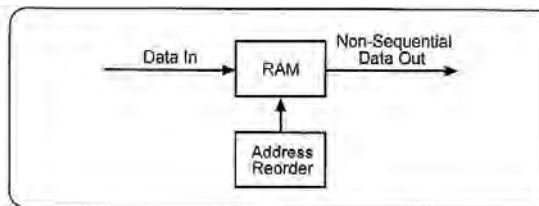
When a burst error reaches the maximum correctable size, the system is vulnerable to random bit errors that make code-words uncorrectable. The use of an inner code applied after interleave and corrected before deinterleave can prevent random errors from entering the deinterleave memory.

As Figure 8-3 shows, when this approach is used with a block interleave structure, the result is a product code. Figure 8-4 shows that interleave can also be convolutional, in which the data array is sheared by applying a different delay to each row. Convolutional, or cross interleave, has the advantage that less memory is needed to interleave and deinterleave.

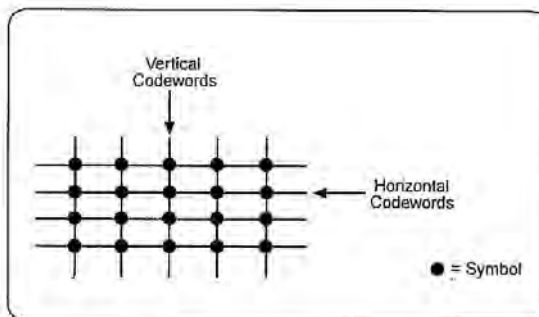
8.5 Channel Coding

Raw-serial binary data is unsuitable for transmission for several reasons. Runs of identical bits cause DC offsets and lack a bit clock. There is no control of the spectrum and the bandwidth required is too great. In practical radio and cable systems, a modulation scheme called a channel code is necessary.

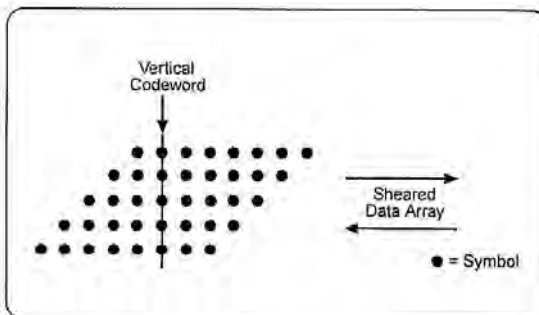
In terrestrial broadcasting, channel bandwidth is at a premium and channel codes which minimize bandwidth are required. In multi-level signaling, the transmitter power is switched between, for example, eight different levels. This approach results in each symbol represents three data bits, requiring one third the bandwidth of binary. As an analog transmitter requires about 8-bit resolution, transmitting three-bit resolution can be achieved with a much lower SNR and with lower transmitter power. In the ATSC system, it is proposed to use a



► Figure 8-2.

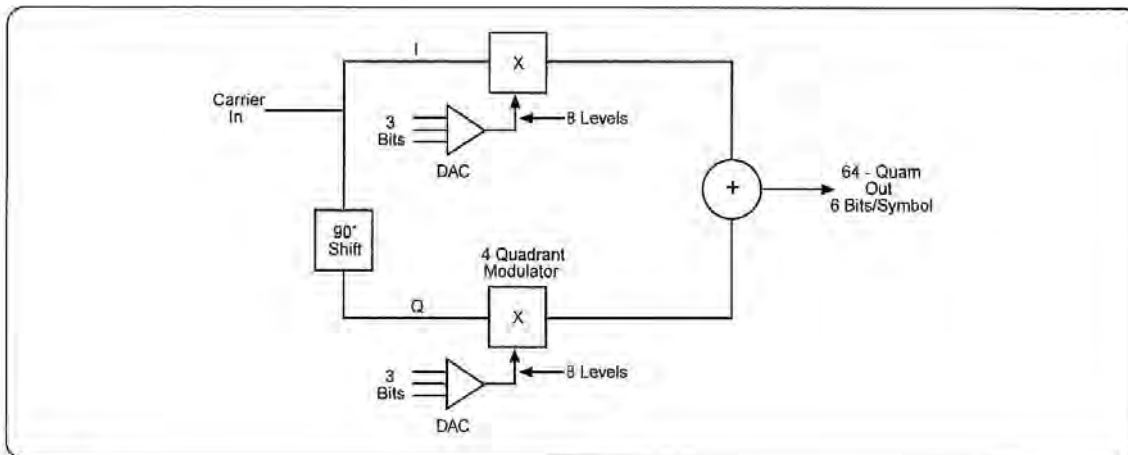


► Figure 8-3.



► Figure 8-4.

system called 8-VSB. Eight-level modulation is used and the transmitter output has a vestigial lower sideband like that used in analog television transmissions. In satellite broadcasting, there is less restriction on bandwidth, but the received signal has a poor SNR, which precludes using multi-level signaling. In this case, it is possible to transmit data by modulating the phase of a signal. In



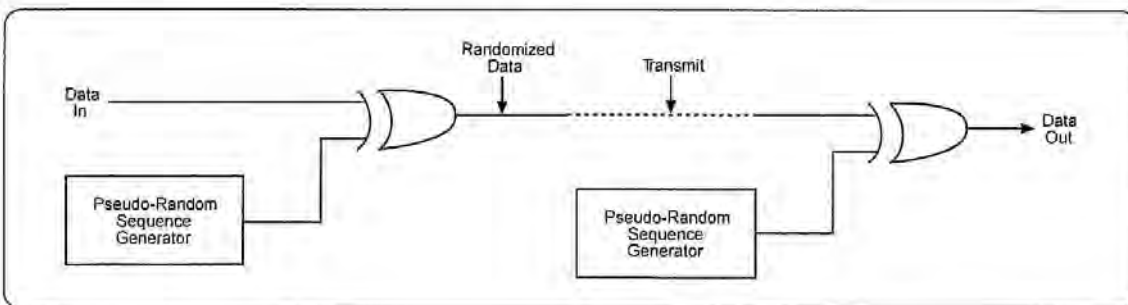
► Figure 8-5.

Phase Shift Keying (PSK), the carrier is transmitted in different phases according to the bit pattern. In Quadrature Phase Shift Keying (QPSK), there are four possible phases. The phases are 90 degrees apart, therefore each symbol carries two bits.

For cable operation, multi-level signaling and phase-shift keying can be combined in Quadrature Amplitude Modulation (QAM), which is a discrete version of the modulation method used in composite video subcarrier. Figure 8-5 shows that if two quadrature carriers, known as I and Q, are individually modulated into eight levels each, the result is a carrier which can have 64 possible combinations of amplitude and phase. Each symbol carries six bits, therefore the bandwidth required is one third that of QPSK.

In the schemes described above, the transmitted signal spectrum is signal dependent. Some parts of the spectrum may contain high energy and cause interference to other services, whereas other parts of the spectrum may contain little energy and be susceptible to interference. In practice, randomizing is necessary to decorrelate the transmitted spectrum from the data content.

Figure 8-6 shows that when randomizing or energy dispersal is used, a pseudo-random sequence is added to the serial data before it is input to the modulator. The result is that the transmitted spectrum is noise-like with relatively stationary statistics. Clearly, an identical and synchronous sequence must be subtracted at the receiver as shown. Randomizing cannot be applied to sync patterns, or they could not be detected.



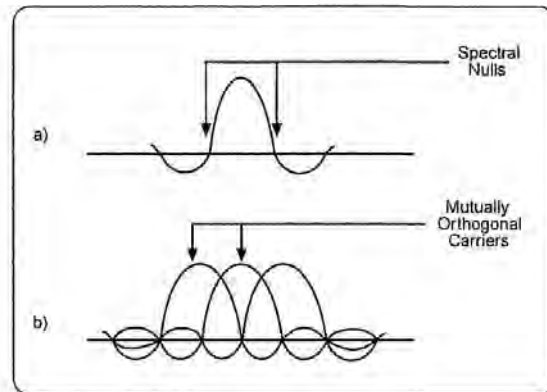
► Figure 8-6.

In the above systems, a baseband signal is supplied to a modulator that operates on a single carrier to produce the transmitted sideband(s). In digital systems with fixed bit rates, the frequency of the sidebands is known. An alternative to a wideband system is one that produces many narrowband carriers at carefully regulated spacing. Figure 8-7a shows that a digitally modulated carrier has a spectral null at each side. Another identical carrier can be placed here without interference because the two are mutually orthogonal as Figure 8-7b shows. This is the principle of COFDM (Coded Orthogonal Frequency Division Multiplexing), which is proposed for use with DVB terrestrial broadcasts (DVB-T).

8.6 Inner Coding

The inner code of a FEC system is designed to prevent random errors from reducing the power of the interleave scheme. A suitable inner code can prevent such errors by giving an apparent increase to the SNR of the transmission. In trellis coding, which can be used with multi-level signaling, several multi-level symbols are associated into a group. The waveform that results from a particular group of symbols is called a trellis. If each symbol can have eight levels, then in three symbols there can be 512 possible trellises.

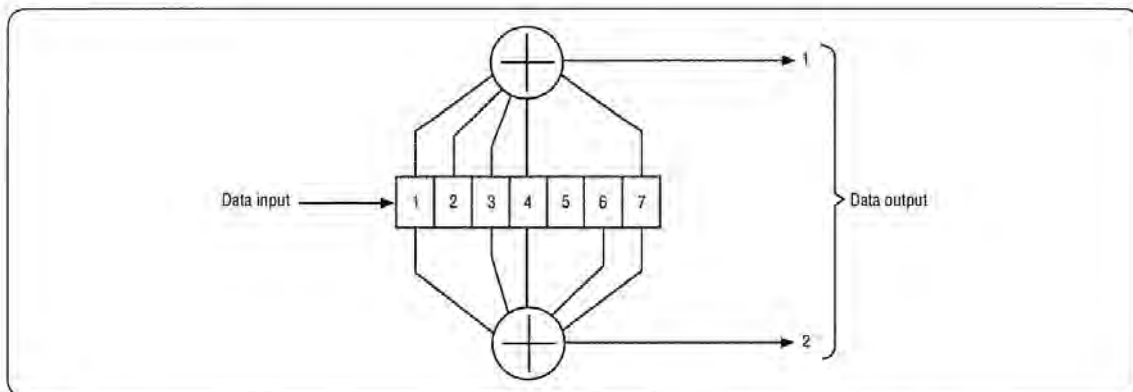
In trellis coding, the data are coded such that only certain trellis waveforms represent valid data. If only 64 of the trellises represent error-free data, then two data bits per symbol can be sent instead of three. The remaining bit is a form of redundancy because trellises other than the correct 64 must be due to errors. If a trellis is received in which the level of one of the symbols is ambiguous due to noise, the ambiguity can be resolved because the correct level must be the one which gives a valid trellis. This technique is known as maximum-likelihood decoding.



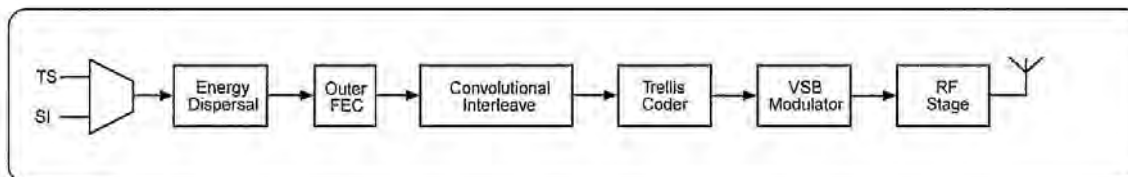
► Figure 8-7.

The 64 valid trellises should be made as different as possible to make the system continue to work with a poorer signal-to-noise ratio. If the trellis coder makes an error, the outer code will correct it.

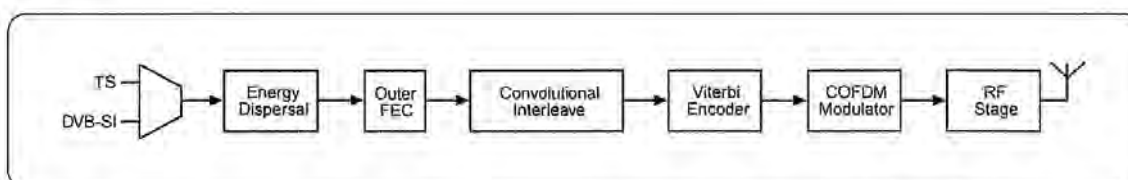
In DVB, Viterbi convolutional coding may be used. Figure 8-8 shows that following interleave, the data are fed to a shift register. The contents of the shift register produce two outputs that represent different parity checks on the input data so that bit errors can be corrected. Clearly, there will be two output bits for every input bit; therefore the coder shown is described as a 1/2 rate coder. Any rate between 1/1 and 1/2 would still allow the original data to be transmitted, but the amount of redundancy would vary. Failing to transmit the entire 1/2



► Figure 8-8.



► Figure 8-9.



► Figure 8-10.

output is called puncturing and it allows any required balance to be obtained between bit rate and correcting power.

8.7 Transmitting Digits

Figure 8-9 shows the elements of an ATSC digital transmitter. Service information describing the transmission is added to the transport stream. This stream is then randomized prior to routing to an outer R-S error correction coder that adds redundancy to the data. A convolutional interleave process then reorders the data so that adjacent data in the transport stream are not adjacent in the transmission. An inner trellis coder is then used to produce a multi-level signal for the VSB modulator.

Figure 8-10 shows a DVB-T transmitter. Service information is added as before, followed by the randomizing stage for energy dispersal. Outer R-S

check symbols are added prior to interleaving. After the interleaver, the inner coding process takes place, and the coded data is fed to a COFDM modulator. The modulator output is then upconverted to produce the RF output.

At the receiver, the bit clock is extracted and used to control the timing of the whole system. The channel coding is reversed to obtain the raw data plus the transmission errors. The inner code corrects random errors and may identify larger errors to help the outer coder after deinterleaving. The randomizing is removed and the result is the original transport stream. The receiver must identify the Program Association Table (PAT) and the Service Information (SI) and Program Map Tables (PMT) that the PAT points to so the viewer can be told what is available in the transport stream and the selected program can be located in the multiplex.

► Section 9 - MPEG Testing

The ability to analyze existing transport streams for compliance is essential, but this ability must be complemented by an ability to create known-compliant transport streams.

9.1 Testing Requirements

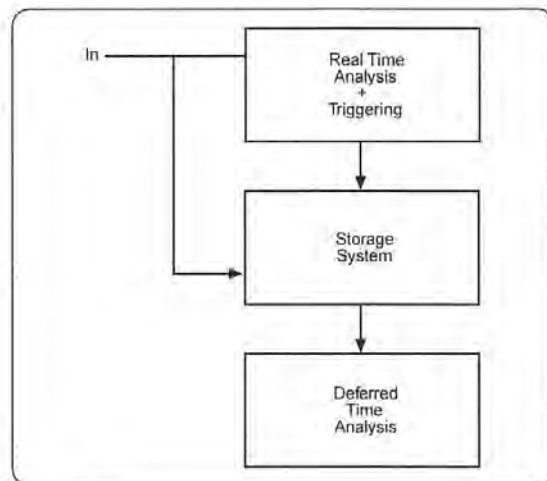
Although the technology of MPEG differs dramatically from the technology that preceded it, the testing requirements are basically the same. On an operational basis, the user wants to have a simple, regular confidence check that ensures all is well. In the event of a failure, the location of the fault needs to be established rapidly. For the purpose of equipment design, the nature of problems needs to be explored in some detail. As with all signal testing, the approach is to combine the generation of known valid signals for insertion into a system with the ability to measure signals at various points.

One of the characteristics of MPEG that distances it most from traditional broadcast video equipment is the existence of multiple information layers, in which each layer is hoped to be transparent to the one below. It is very important to be able to establish in which layer any fault resides to avoid a fruitless search.

For example, if the picture monitor on an MPEG decoder is showing visible defects, these defects could be due to a number of possibilities. Perhaps the encoder is faulty, and the transport stream is faithfully delivering the faulty information. On the other hand, the encoder might be fine, but the transport layer is corrupting the data. In DVB, there are even more layers such as energy dispersal, error correction, and interleaving. Such complexity requires a structured approach to fault finding, using the right tools. The discussion of protocol analysis of the compressed data in this primer may help the user derive such an approach. Reading the discussion of another important aspect of testing for compressed television, picture-quality assessment, may also be helpful. This later discussion is found in the Tektronix publication, *A Guide to Video Measurements for Compressed Television Systems*.

9.2 Analyzing a Transport Stream

An MPEG transport stream has an extremely complex structure, but an analyzer such as the MTS 100 can break down the structure in a logical fashion such that the user can observe any required details. Many general types of analysis can take place in real time on a live transport stream. These include displays of the hierarchy of programs in the transport stream and of the proportion of the stream bit rate allocated to each stream.



► Figure 9-1.

More detailed analysis is only possible if part of a transport stream is recorded so that it can be picked apart later. This technique is known as deferred-time testing and could be used, for example, to examine the contents of a time stamp.

When used for deferred-time testing, the MPEG transport-stream analyzer is acting like a logic analyzer that provides data-interpretation tools specific to MPEG. Like all logic analyzers, a real-time triggering mechanism is required to determine the time or conditions under which capture will take place. Figure 9-1 shows that an analyzer contains a real-time section, a storage section, and a deferred section. In real-time analysis, only the real-time section operates, and a signal source needs to be connected. For capture, the real-time section is used to determine when to trigger the capture. The analyzer includes tools known as filters that allow selective analysis to be applied before or after capture.

Once the capture is completed, the deferred section can operate on the captured data and the input signal is no longer necessary. There is a good parallel in the storage oscilloscope which can display the real-time input directly or save it for later study.

A Guide to MPEG Fundamentals and Protocol Analysis

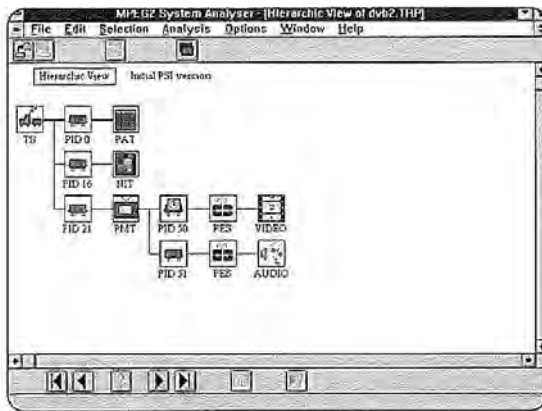


Figure 9-2.

9.3 Hierarchic View

When analyzing an unfamiliar transport stream, the hierarchic view is an excellent starting point because it enables a graphic view of every component in the stream. Figure 9-2 shows an example of a hierarchic display such as that provided by the Tektronix MTS 200. Beginning at top left of the entire transport stream, the stream splits and an icon is presented for every stream component. Table 9-1 shows the different icons that the hierarchical view uses and their meaning. The user can very easily see how many program streams are present and the video and audio content of each. Each icon represents the top layer of a number of lower analysis and information layers.

The analyzer creates the hierarchic view by using the PAT and PMT in the Program Specific Information (PSI) data in the transport stream. The PIDs from these tables are displayed beneath each icon. PAT and PMT data are fundamental to the operation of any demultiplexer or decoder; if the analyzer cannot display a hierarchic view or displays a view which is obviously wrong, the transport stream under test has a PAT/PMT error. It is unlikely that equipment further up the line will be able to interpret the stream at all.

The ability of a demux or decoder to lock to a transport stream depends on the frequency with which the PSI data are sent. The PSI/SI rate option shown in



Figure 9-3.

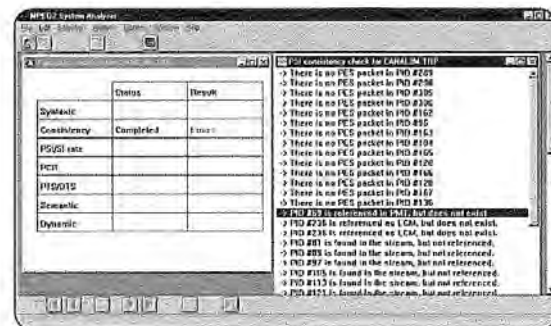


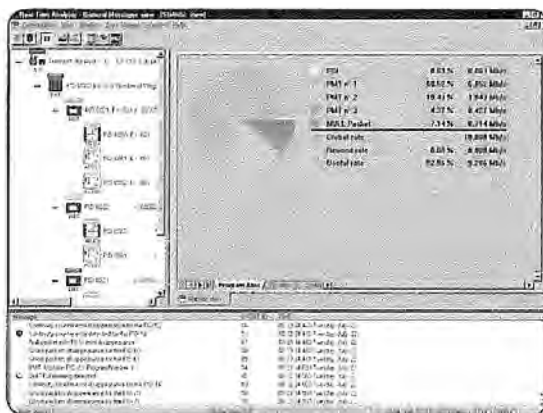
Figure 9-4.

Figure 9-3 displays the frequency of insertion of system information. PSI/SI information should also be consistent with the actual content in the bit stream. For example, if a given PID is referenced in a PMT, it should be possible to find PIDs of this value in the bit stream. The consistency-check function makes such a comparison. Figure 9-4 shows a consistency-error readout.

► Table 9-1. Hierarchical View Icons

Icon	Element Type
	Multiplex transport packets. This icon represents all (188- and 204-byte transport packets) that make up the stream. If you visualize the transport stream as a train, this icon represents every car in the train, regardless of its configuration (for example, flat car, boxcar, or hopper) and what it contains.
	Transport packets of a particular PID (Program ID). Other elements (tables, clocks, PES packets) are the "payload" contained within transport packets or are constructed from the payload of several transport packets that have the same PID. The PID number appears under the icon.
	Transport Packets that contain independent PCR clocks. The PID appears under the icon.
	PAT (Program Association Table) sections. Always contained in PID 0 transport packets.
	PMT (Program Map Table) sections.
	NIT (Network Information Table). Provides access to SI Tables through the PSI/SI command from the Selection menu. Also used for Private sections. When the DVB option (in the Options menu) is selected, this icon can also represent SDT, BAT, EIT, and TDT sections.
	Packetized Elementary Stream (PES). This icon represents all packets that, together, contain a given elementary stream. Individual PES packets are assembled from the payloads of several transport packets.
	Video elementary stream.
	Audio elementary stream.
	Data elementary stream.
	ECM (Entitlement Control Message) sections.
	EMM (Entitlement Management Message) sections.

A Guide to MPEG Fundamentals and Protocol Analysis



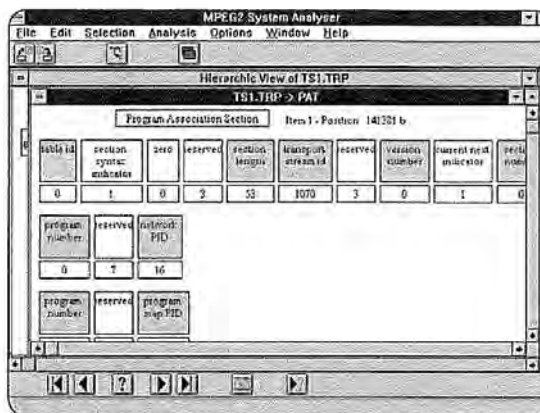
► Figure 9-5.

A MUX allocation chart may graphically display the proportions of the transport stream allocated to each PID or program. Figure 9-5 shows an example of a MUX-allocation pie-chart display. The hierarchical view and the MUX Allocation Chart show the number of elements in the transport stream and the proportion of bandwidth allocated, but they do not show how the different elements are distributed in time in the multiplex. The PID map function displays, in order of reception, a list of the PID of every packet in a section of the transport stream. Each PID type is color-coded differently, so it is possible to assess the uniformity of the multiplexing. Bursts of SI data in a Transport Stream, sent as contiguous packets, may cause buffer overflows in the decoder.

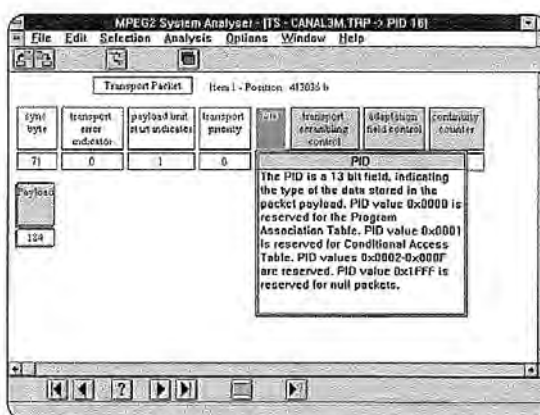
9.4 Interpreted View

As an alternative to checking for specific data in unspecified places, it is possible to analyze unspecified data in specific places, including in the individual transport stream packets, the tables, or the PES packets. This analysis is known as the interpreted view because the analyzer automatically parses and decodes the data and then displays its meaning. Figure 9-6 shows an example of an interpreted view. As the selected item is changed, the elapsed time and byte count relative to the start of the stream can be displayed.

The interpreted view can also be expanded to give a full explanation of the meaning of any field in the interpreted item. Figure 9-7 shows an example of a field explanation. Because MPEG has so many different parameters, field explanations can help recall their meanings.



► Figure 9-6.



► Figure 9-7.

9.5 Syntax and CRC Analysis

To ship program material, the transport stream relies completely on the accurate use of syntax by encoders. Without correct settings of fixed flag bits, sync patterns, packet-start codes, and packet counts, a decoder may misinterpret the bit stream. The syntax check function considers all bits that are not program material and displays any discrepancies. Spurious discrepancies could be due to transmission errors; consistent discrepancies point to a faulty encoder or multiplexer. Figure 9-8 shows a syntax-error display.

Many MPEG tables have checksums or CRCs attached for error detection. The analyzer can recalculate the checksums and compare them with the actual checksum. Again, spurious CRC mismatches could be due to stream-bit errors, but consistent CRC errors point to a hardware fault.

9.6 Filtering

A transport stream contains a great amount of data, and in real fault conditions, it is probable that, unless a serious problem exists, much of the data is valid and that perhaps only one elementary stream or one program is affected. In this case, it is more effective to test selectively, which is the function of filtering.

Essentially, filtering allows the user of an analyzer to be more selective when examining a transport stream. Instead of accepting every bit, the user can analyze only those parts of the data that meet certain conditions.

One condition results from filtering packet headers so that only packets with a given PID are analyzed. This approach makes it very easy to check the PAT by selecting PID 0, and, from there, all other PIDs can be read out. If the PIDs of a suspect stream are known, perhaps from viewing a hierarchical display, it is easy to select a single PID for analysis.

Alternatively, the filtering can be used to search for unknown PIDs on the basis of some condition. For example, if it is not known what audio compression levels are being used, it is possible to search audio sequence headers containing, say, Layer 2 coding.

It may be useful to filter so that only certain elementary-stream access units are selected. At a very low level, it may be useful to filter on a specific four-byte bit pattern. In the case of false detection of sync patterns, this filter would allow the source of the false syncs to be located.

In practice, it is useful to be able to concatenate filters. In other words, it is possible to filter not just on packets having a PID of a certain value, but perhaps only packets of that PID value which also contain a PES header. By combining filters in this way, it is possible to be very specific about the data to be displayed. As fault finding proceeds, this progressive filtering allows optimization of the search process.

9.7 Timing Analysis

The tests described above check for the presence of the correct elements and syntax in the transport stream. However, to display real-time audio and video correctly, the transport stream must also deliver accurate timing to the

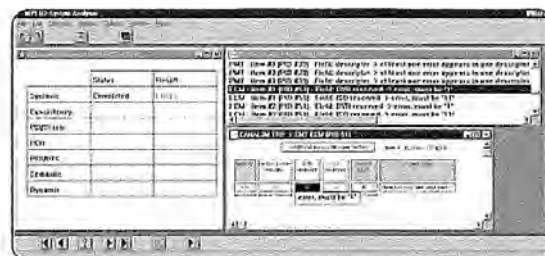


Figure 9-8.

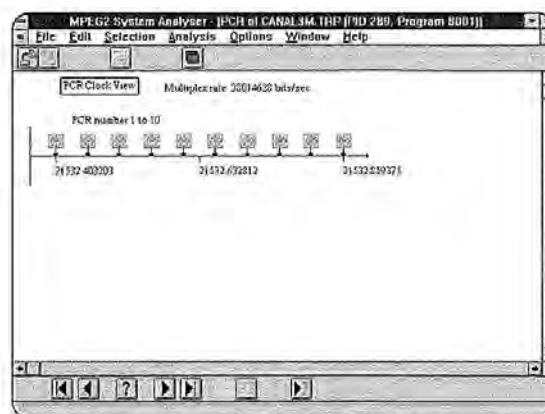


Figure 9-9.

decoders. This task can be confirmed by analyzing the PCR and time-stamp data.

The correct transfer of program-clock data is vital because this data controls the entire timing of the decoding process. PCR analysis can show that, in each program, PCR data is sent at a sufficient rate and with sufficient accuracy to be compliant.

The PCR data from a multiplexer may be precise, but remultiplexing may put the packets of a given program at a different place on the time axis, requiring that the PCR data be edited by the remultiplexer. Consequently, it is important to test for PCR jitter after the data is remultiplexed.

Figure 9-9 shows a PCR display that indicates the times at which PCRs were received. At the next display level, each PCR can be opened to display the PCR

A Guide to MPEG Fundamentals and Protocol Analysis

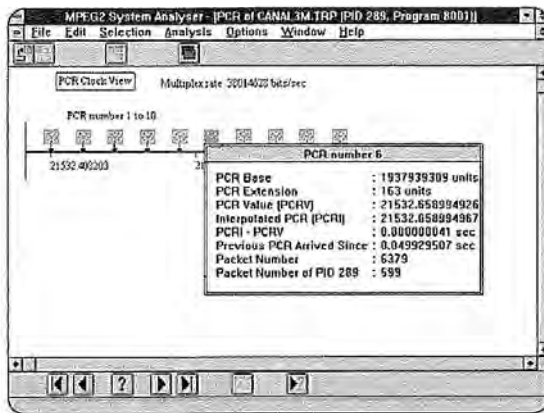


Figure 9-10.

data, as is shown in Figure 9-10. To measure jitter, the analyzer predicts the PCR value by using the previous PCR and the bit rate to produce what is called the interpolated PCR or PCRI. The actual PCR value is subtracted from PCRI to give an estimate of the jitter. The figure also shows the time since the previous PCR arrived.

An alternate approach shown in Figure 9-11 provides a graphical display of PCR jitter and PCR repetition rate which is updated in real-time.

Once PCR data is known to be correct, the time stamps can be analyzed. Figure 9-12 shows a time-stamp display for a selected elementary stream.

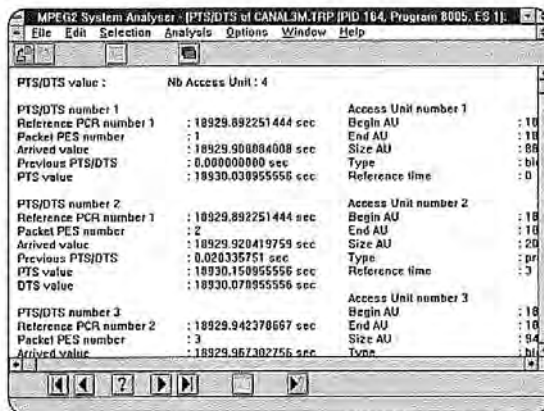


Figure 9-12.

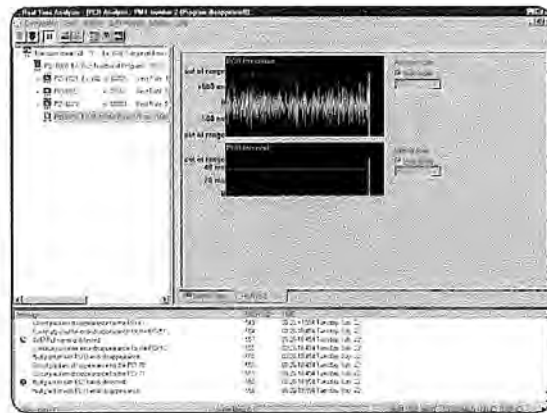


Figure 9-11.

time of arrival, the presentation time, and, where appropriate, the decode times are all shown.

In MPEG, the reordering and use of different picture types causes delay and requires buffering at both encoder and decoder. A given elementary stream must be encoded within the constraints of the availability of buffering at the decoder. MPEG defines a model decoder called the T-STD (Transport Stream Target Decoder); an encoder or multiplexer must not distort the data flow beyond the buffering ability of the T-STD. The transport stream contains parameters called VBV (Video Buffer Verify) specifying the amount of buffering needed by a given elementary stream.

The T-STD analysis displays the buffer occupancy graphically so that overflows or underflows can be easily seen. Figure 9-13 shows a buffering display.

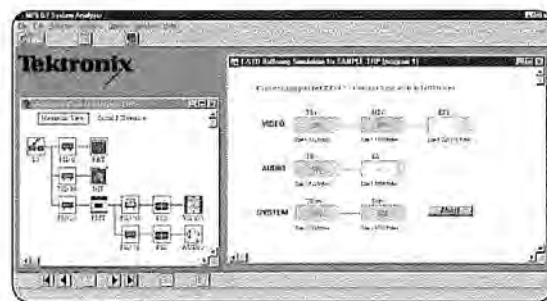


Figure 9-13.

The output of a normal compressor/multiplexer is of limited use because it is not deterministic. If a decoder defect is seen, there is no guarantee that the same defect will be seen on a repeat of the test because the same video signal will not result in the same transport stream. In this case, an absolutely repeatable transport stream is essential so that the defect can be made to occur at will for study or rectification.

Transport stream jitter should be within certain limits, but a well-designed decoder should be able to recover programs beyond this limit in order to guarantee reliable operation. There is no way to test for this capability using existing transport streams because, if they are compliant, the decoder is not being tested. If there is a failure, it will not be reproducible and it may not be clear whether the failure was due to jitter or some other noncompliance. The solution is to generate a transport stream that is compliant in every respect and then add a controlled amount of jitter to it so that jitter is then known to be the only source of noncompliance. The multiplexer feature of the MTS 200 is designed to create such signals.

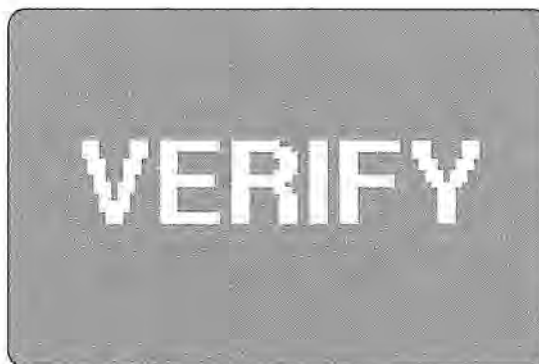
9.8 Elementary Stream Testing

Because of the flexible nature of the MPEG bit stream, the number of possibilities and combinations it can contain is almost incalculable. As the encoder is not defined, encoder manufacturers are not compelled to use every possibility; indeed, for economic reasons, this is unlikely. This fact makes testing quite difficult because the fact that a decoder works with a particular encoder does not prove compliance. That decoder may simply not be using the modes that cause the decoder to fail.

A further complication occurs because encoders are not deterministic and will not produce the same bit stream if the video or audio input is repeated. There is little chance that the same alignment will exist between I, P, and B pictures and the video frames. If a decoder fails a given test, it may not fail the next time the test is run, making fault-finding difficult. A failure with a given encoder does not determine whether the fault lies with the encoder or the decoder. The coding difficulty depends heavily on the nature of the program material, and any given program material will not necessarily exercise every parameter over the whole coding range.

To make tests that have meaningful results, two tools are required:

- ▶ A known source of compliant test signals that deliberately explore the whole coding range. These signals must be deterministic so that a decoder failure will give repeatable symptoms. The Sarnoff compliant bit streams are designed to perform this task.
- ▶ An elementary stream analyzer that allows the entire syntax from an encoder to be checked for compliance.



▶ Figure 9-14.

9.9 Sarnoff Compliant Bit Streams

These bit streams have been specifically designed by the David Sarnoff Research Center for decoder compliance testing. They can be multiplexed into a transport stream feeding a decoder.

No access to the internal working of the decoder is required. To avoid the need for lengthy analysis of the decoder output, the bit streams have been designed to create a plain picture when they complete so that it is only necessary to connect a picture monitor to the decoder output to view them.

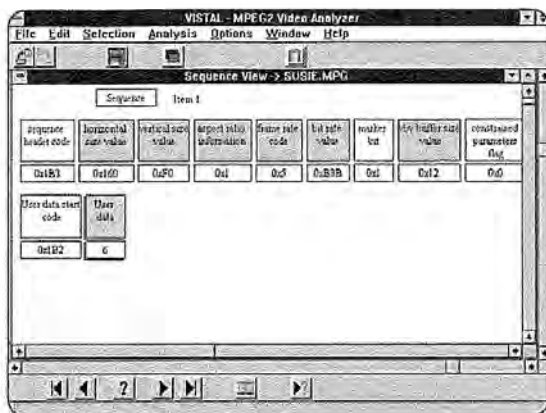
There are a number of these simple pictures. Figure 9-14 shows the gray verify screen. The user should examine the verify screen to look for discrepancies that will display well against the gray field. There are also some verify pictures which are not gray.

Some tests will result in no picture at all if there is a failure. These tests display the word "VERIFY" on screen when they complete.

Further tests require the viewer to check for smooth motion of a moving element across the picture. Timing or ordering problems will cause visible jitter.

The suite of Sarnoff tests may be used to check all of the MPEG syntax elements in turn. In one test, the bit stream begins with I pictures only, adds P pictures, and then adds B pictures to test whether all MPEG picture types can be handled and correctly reordered. Backward compatibility with MPEG-1 can be proven. Another bit-stream tests using a range of different GOP structures. There are tests that check the operation of motion vectors over the whole range of values, and there are tests that vary the size of slices or the amount of stuffing.

A Guide to MPEG Fundamentals and Protocol Analysis



► Figure 9-15.

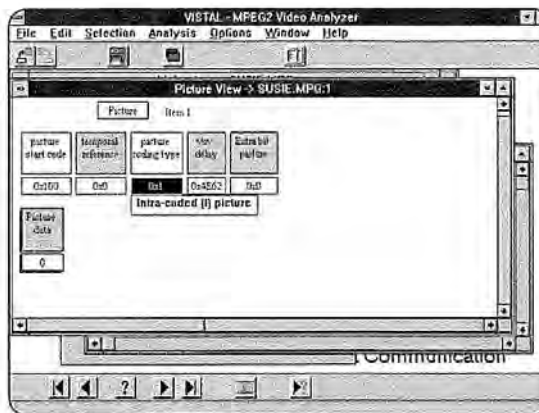
In addition to providing decoder tests, the Sarnoff streams also include sequences that cause a good decoder to produce standard video test signals to check DACs, signal levels, and composite or Y/C encoders. These sequences turn the decoder into a video test-pattern generator capable of producing conventional video signals such as zone plates, ramps, and color bars.

9.10 Elementary Stream Analysis

An elementary stream is a payload that the transport stream must deliver transparently. The transport stream will do so whether or not the elementary stream is compliant. In other words, testing a transport stream for compliance simply means checking that it is delivering elementary streams unchanged. It does not mean that the elementary streams were properly assembled in the first place.

The elementary-stream structure or syntax is the responsibility of the compressor. Therefore an elementary stream test is essentially a form of compressor test. It should be noted that a compressor can produce compliant syntax, and yet still have poor audio or video quality. However, if the syntax is incorrect, a decoder may not be able to interpret the elementary stream. Since compressors are algorithmic rather than deterministic, an elementary stream may be intermittently noncompliant if some less common mode of operation is not properly implemented.

As transport streams often contain several programs that come from different coders, elementary-stream problems tend to be restricted to one program, whereas transport stream problems tend to affect all programs. If problems are noted with the output of a particular decoder, then the Sarnoff compliance tests



► Figure 9-16.

should be run on that decoder. If these are satisfactory, the fault may lie in the input signal. If the transport stream syntax has been tested, or if other programs are working without fault, then an elementary stream analysis is justified.

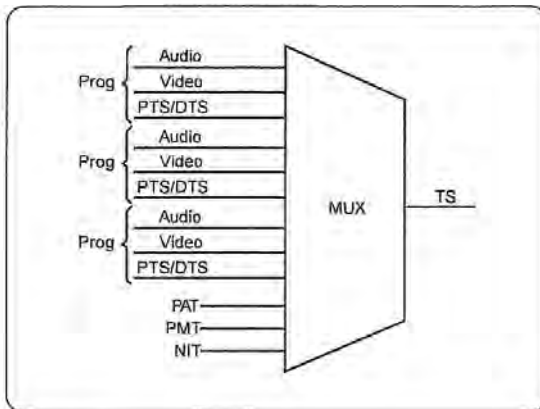
Elementary stream analysis can begin at the top level of the syntax and continue downwards. Sequence headers are very important as they tell the decoder all of the relevant modes and parameters used in the compression. The elementary-stream syntax described in Sections 4.1 and 4.2 should be used as a guide. Figure 9-15 shows part of a sequence header displayed on an MTS 200. At a lower level of syntax, Figure 9-16 shows a picture header.

9.11 Creating a Transport Stream

Whenever the decoder is suspect, it is useful to be able to generate a test signal of known quality. Figure 9-17 shows that an MPEG transport stream must include Program Specific Information (PSI), such as PAT, PMT, and NIT describing one or more program streams. Each program stream must contain its own program clock reference (PCR) and elementary streams having periodic time stamps.

A DVB transport stream will contain additional service information, such as BAT, SDT, and EIT tables. A PSI/SI editor enables insertion of any desired compliant combination of PSI/SI into a custom test stream.

Clearly, each item requires a share of the available transport-stream rate. The multiplexer provides a rate gauge to display the total bit rate used. The remainder of the bit rate is used up by inserting stuffing packets with PIDs that contain all 1s, which a decoder will reject.



► Figure 9-17.

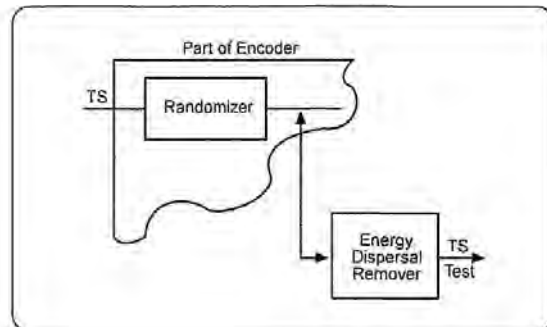
9.12 Jitter Generation

The MPEG decoder has to recreate a continuous clock by using the clock samples in PCR data to drive a phase-locked loop. The loop needs filtering and damping so that jitter in the time of arrival of PCR data does not cause instability in the clock.

To test the phase-locked loop performance, a signal with known jitter is required; otherwise, the test is meaningless. The MTS 200 can generate simulated jitter for this purpose. Because it is a reference generator, the MTS 200 has highly stable clock circuits and the actual output jitter is very small. To create the effect of jitter, the timing of the PCR data is not changed at all. Instead, the PCR values are modified so that the PCR count they contain is slightly different from the ideal. The modified value results in phase errors at the decoder that are indistinguishable from real jitter.

The advantage of this approach is that jitter of any required magnitude can easily be added to any program stream simply by modifying the PCR data and leaving all other data intact. Other program streams in the transport stream need not have jitter added. In fact, it may be best to have a stable program stream to use as a reference.

For different test purposes, the time base may be modulated in a number of ways that determine the spectrum of the loop phase error in order to test the loop filtering. Square-wave jitter alternates between values which are equally early or late. Sinusoidal jitter values cause the phase error to be a sampled sine wave. Random jitter causes the phase error to be similar to noise.



► Figure 9-18.

The MTS 200 also has the ability to add a fixed offset to the PCR data. This offset will cause a constant shift in the decode time. If two programs in the same transport stream are decoded to baseband video, a PCR shift in one program produces a change of sync phase with respect to the unmodified program.

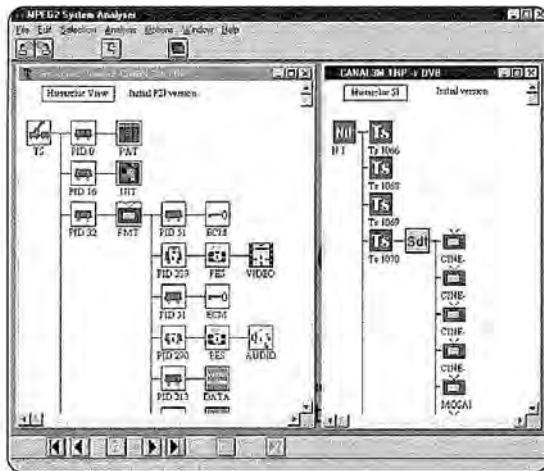
9.13 DVB Tests

DVB transmission testing can be divided into distinct areas. The purpose of the DVB equipment (FEC coder, transmitter, receiver, and error correction) is to deliver a transport stream to the receiver with negligible error. In this sense, the DVB layer is (or should be) transparent to MPEG, and certain testing can assume that the DVB transmission layer is intact. This assumption is reasonable because any defect in the DVB layer will usually result in excessive bit errors, and this condition will cause both an MPEG decoder and a syntax tester to misinterpret the stream. Consequently, if a transport stream tester connected to a DVB/ATSC receiver finds a wide range of many errors, the problem is probably not in the MPEG equipment, but due to errors in the transmission.

Finding transmission faults requires the ability to analyze various points in the DVB Chain, checking for conformance with DVB specifications and MPEG standards. Figure 8-1 showed the major processes in the DVB chain. For example, if an encoder and an IRD (integrated receiver decoder) are incompatible, which one is in error? Or, if the energy dispersal randomizing is not correct at either encoder or decoder, how is one to proceed?

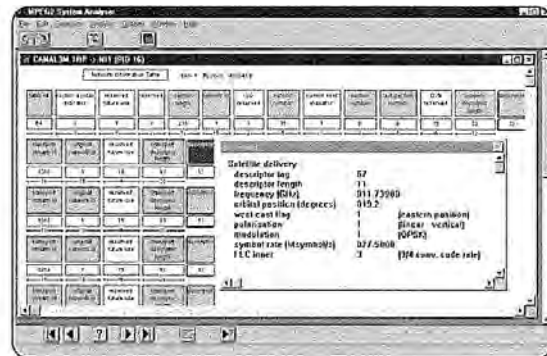
The solution is to use an analyzer that can add or remove compliant energy dispersal. Figure 9-18 shows that if energy dispersal is removed, the encoder randomizer can be tested. Alternatively, if randomized data are available, the energy dispersal removal at the IRD can be tested.

A Guide to MPEG Fundamentals and Protocol Analysis



► Figure 9-19.

Similarly, the MTS 200 can add and check Reed-Solomon protection and can interleave and deinterleave data. These stages in the encoder and the reverse stages in the decoder can be tested. The inner redundancy after interleaving can also be added and checked. The only process the MTS 200 cannot perform is the modulation, which is an analog process. However, the MTS 100 can drive a modulator with a compliant signal so that it can be tested on a constellation analyzer or with an IRD. This approach is also a very simple way of producing compliant test transmissions.



► Figure 9-20.

Once the DVB layer is found to be operating at an acceptable error rate, a transport stream tester at the receiver is essentially testing the transport stream as it left the encoder or remultiplexer. Therefore, the transport-stream test should be performed at a point prior to the transmitter. The transport stream tests should be performed using the tests outlined above. Most of these tests operate on transport stream headers, and so there should be no difficulty if the program material is scrambled.

In addition to these tests, the DVB-specific tests include testing for the presence and the frequency of the SDT (Service Description Table), NIT (Network Information Table), and the EIT (Event Information Table). This testing can be performed using the DVB option in the MTS 200. Figure 9-19 shows a hierarchic view of DVB-SI tables from a transport stream. Figure 9-20 shows a display of decoded descriptor data.

► Glossary

AAU – Audio Access Unit. See Access unit.

AC-3 – An audio-coding technique used with ATSC (see *Section 4 – Elementary Streams*).

Access Unit – The coded data for a picture or block of sound and any stuffing (null values) that follows it.

ATSC – Advanced Television Systems Committee.

Bouquet – A group of transport streams in which programs are identified by combination of network ID and PID (part of DVB-SI).

CAT – Conditional Access Table. Packets having PID (see *Section 7 – Transport Streams*) codes of 1 and that contain information about the scrambling system. See ECM and EMM.

Channel Code – A modulation technique that converts raw data into a signal that can be recorded or transmitted by radio or cable.

CIF – Common Interchange Format. A 352 x 240 pixel format for 30 fps video conferencing.

Closed GOP – A Group Of Pictures in which the last pictures do not need data from the next GOP for bidirectional coding. Closed GOP is used to make a splice point in a bit stream.

Coefficient – A number specifying the amplitude of a particular frequency in a transform.

DCT – Discrete Cosine Transform.

DTS – Decoding Time Stamp. Part of PES header indicating when an access unit is to be decoded.

DVB – Digital Video Broadcasting. The broadcasting of television programs using digital modulation of a radio frequency carrier. The broadcast can be terrestrial or from a satellite.

DVB-IRD – See IRD.

DVB-SI – DVB Service Information. Information carried in a DVB multiplex describing the contents of different multiplexes. Includes NIT, SDT, EIT, TDT, BAT, RST, and ST (see *Section 8 – Introduction to DVB/ATSC*).

DVC – Digital Video Cassette.

Elementary Stream – The raw output of a compressor carrying a single video or audio signal.

ECM – Entitlement Control Message. Conditional access information specifying control words or other stream-specific scrambling parameters.

EIT – Event Information Table. Part of DVB-SI.

EMM – Entitlement Management Message. Conditional access information specifying authorization level or services of specific decoders. An individual decoder or a group of decoders may be addressed.

ENG – Electronic News Gathering. Term used to describe use of video-recording instead of film in news coverage.

EPG – Electronic Program Guide. A program guide delivered by data transfer rather than printed paper.

FEC – Forward Error Correction. System in which redundancy is added to the message so that errors can be corrected dynamically at the receiver.

GOP – Group Of Pictures. A GOP starts with an I picture and ends with the last picture before the next I picture.

Inter-coding – Compression that uses redundancy between successive pictures; also known as temporal coding.

Interleaving – A technique used with error correction that breaks up burst errors into many smaller errors.

Intra-coding – Compression that works entirely within one picture; also known as spatial coding.

IRD – Integrated Receiver Decoder. A combined RF receiver and MPEG decoder that is used to adapt a TV set to digital transmissions.

Level – The size of the input picture in use with a given profile (see *Section 2 – Compression in Video*).

Macroblock – The screen area represented by several luminance and color-difference DCT blocks that are all steered by one motion vector.

Masking – A psycho-acoustic phenomenon whereby certain sounds cannot be heard in the presence of others.

NIT – Network Information Table. Information in one transport stream that describes many transport streams.

Null Packets – Packets of “stuffing” that carry no data but are necessary to maintain a constant bit rate with a variable payload. Null packets always have a PID of 8191 (all 1s). (See *Section 7 – Transport Streams*).

PAT – Program Association Table. Data appearing in packets having PID (see *Section 7 – Transport Streams*) code of zero that the MPEG decoder uses to determine which programs exist in a Transport Stream. PAT points to PMT, which, in turn, points to the video, audio, and data content of each program.

A Guide to MPEG Fundamentals and Protocol Analysis

PCM – Pulse Code Modulation. A technical term for an analog source waveform, for example, audio or video signals, expressed as periodic, numerical samples. PCM is an uncompressed digital signal.

PCR – Program Clock Reference. The sample of the encoder clock count that is sent in the program header to synchronize the decoder clock.

PCRI – Interpolated Program Clock Reference. A PCR estimated from a previous PCR and used to measure jitter.

PID – Program Identifier. A 13-bit code in the transport packet header. PID 0 indicates that the packet contains a PAT PID. (See **Section 7 – Transport Streams**.) PID 1 indicates a packet that contains CAT. The PID 8191 (all 1s) indicates null (stuffing) packets. All packets belonging to the same elementary stream have the same PID.

PMT – Program Map Tables. The tables in PAT that point to video, audio, and data content of a transport stream.

Packets – A term used in two contexts: in program streams, a packet is a unit that contains one or more presentation units; in transport streams, a packet is a small, fixed-size data quantum.

Preprocessing – The video signal processing that occurs before MPEG Encoding. Noise reduction, downsampling, cut-edit identification, and 3:2 pull-down identification are examples of preprocessing.

Profile – Specifies the coding syntax used.

Program Stream – A bit stream containing compressed video, audio, and timing information.

PTS – Presentation Time Stamps. The time at which a presentation unit is to be available to the viewer.

PSI – Program Specific Information. Information that keeps track of the different programs in an MPEG transport stream and in the elementary streams in each program. PSI includes PAT, PMT, NIT, CAT, ECM, and EMM.

PSI/SI – A general term for combined MPEG PSI and DVB-SI.

PU – Presentation Unit. One compressed picture or block of audio.

QCIF – One-quarter-resolution (176 x 144 pixels) Common Interchange Format. See CIF.

QSIF – One-quarter-resolution Source Input Format. See SIF.

RLC – Run Length Coding. A coding scheme that counts number of similar bits instead of sending them individually.

SDI – Serial Digital Interface. Serial coaxial cable interface standard intended for production digital video signals.

STC – System Time Clock. The common clock used to encode video and audio in the same program.

SDT – Service Description Table. A table listing the providers of each service in a transport stream.

SI – See DVB-SI.

SIF – Source Input Format. A half-resolution input signal used by MPEG-1.

ST – Stuffing Table.

Scalability – A characteristic of MPEG-2 that provides for multiple quality levels by providing layers of video data. Multiple layers of data allow a complex decoder to produce a better picture by using more layers of data, while a more simple decoder can still produce a picture using only the first layer of data.

Stuffing – Meaningless data added to maintain constant bit rate.

Syndrome – Initial result of an error checking calculation. Generally, if the syndrome is zero, there is assumed to be no error.

TDAC – Time Domain Aliasing Cancellation. A coding technique used in AC-3 audio compression.

TDT – Time and Date Table. Used in DVB-SI.

T-STD – Transport Stream System Target Decoder. A decoder having a certain amount of buffer memory assumed to be present by an encoder.

Transport Stream – A multiplex of several program streams that are carried in packets. Demultiplexing is achieved by different packet IDs (PIDs). See PSI, PAT, PMT, and PCR.

Truncation – Shortening the wordlength of a sample or coefficient by removing low-order bits.

VAU – Video Access Unit. One compressed picture in program stream.

VLC – Variable Length Coding. A compressed technique that allocates short codes to frequency values and long codes to infrequent values.

VOD – Video On Demand. A system in which television programs or movies are transmitted to a single consumer only when requested.

Vector – A motion compensation parameter that tells a decoder how to shift part of a previous picture to more closely approximate the current picture.

Wavelet – A transform in the basis function that is not of fixed length but that grows longer as frequency reduces.

Weighting – A method of changing the distribution of the noise that is due to truncation by pre-multiplying values.

www.tektronix.com/video_audio/

ASEAN Countries (65) 356-3900

Australia & New Zealand 61 (2) 9888-0100

Austria, Central Eastern Europe

Greece, Turkey, Malta & Cyprus +43 2236 8092 0

Belgium +32 (2) 715 89 70

Brazil and South America 55 (11) 3741-8360

Canada 1 (800) 661-5625

Denmark +45 (44) 850 700

Finland +358 (9) 4783 400

France & North Africa +33 1 69 86 81 81

Germany +49 (221) 94 77 400

Hong Kong (852) 2585-6688

India (91) 80-2275577

Italy +39 (2) 25086 501

Japan (Sony/Tektronix Corporation) 81 (3) 3448-3111

Mexico, Central America, & Caribbean 52 (5) 666-6333

The Netherlands +31 23 56 95555

Norway +47 22 07 07 00

People's Republic of China 86 (10) 6235 1230

Republic of Korea 82 (2) 528-5289

South Africa (27 11) 651-5222

Spain & Portugal +34 91 372 6000

Sweden +46 8 477 65 00

Switzerland +41 (41) 729 36 40

Taiwan 886 (2) 2722-9822

United Kingdom & Eire +44 (0)1344 392000

USA 1 (800) 426-2200

For other areas, contact Tektronix, Inc. at 1 (503) 627-1924

For Further Information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs, and other resources to help engineers working on the cutting edge of technology.

Please visit "Resources For You" on our Web site at www.tektronix.com



Copyright © 2000, Tektronix, Inc. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

1800-109686 1888-114444



Tektronix

Ex. 1014A



Video Test Equipment

[Video Test Home](#)

Product Portfolio

[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Related Products](#)
[Product Finder](#)

Resources For You

[Technical Papers](#)
[Manuals](#)
[Support Tools](#)
[Service](#)
[Software & Drivers](#)
[RGB Newsletter](#)
[Ask the Experts](#)

What's New

None



A Guide to MPEG Fundamentals and Protocol Analysis

Including DVB and ATSC

Contents

[Print Application Note](#) (PDF)

[Section 1 Introduction to MPEG](#)

- [1.1 Convergence](#)
- [1.2 Why compression is needed](#)
- [1.3 Applications of compression](#)
- [1.4 Introduction to video compression](#)
- [1.5 Introduction to audio compression](#)



Video Test Equipment

[▶ Video Test Home](#)

Product Portfolio

- [Video Measurement](#)
- [Signal Generators](#)
- [Signal Monitors](#)
- [MPEG Test](#)
- [Picture Quality](#)
- [RF Modulation](#)
- [Audio Test](#)
- [Related Products](#)
- [Product Finder](#)

Resources For You

- [Technical Papers](#)
- [Manuals](#)
- [Support Tools](#)
- [Service](#)
- [Software & Drivers](#)
- [RGB Newsletter](#)
- [Ask the Experts](#)

What's New

- [News](#)

- [1.5 Introduction to audio compression](#)
- [1.6 MPEG signals](#)
- [1.7 Need for monitoring and analysis](#)
- [1.8 Pitfalls of compression](#)

Section 2 Compression in Video

- [2.1 Spatial or temporal coding?](#)
- [2.2 Spatial coding](#)
- [2.3 Weighting](#)
- [2.4 Scanning](#)
- [2.5 Entropy coding](#)
- [2.6 A spatial coder](#)
- [2.7 Temporal coding](#)
- [2.8 Motion compensation](#)
- [2.9 Bidirectional coding](#)
- [2.10 I, P, and B pictures](#)
- [2.11 An MPEG compressor](#)
- [2.12 Preprocessing](#)
- [2.13 Profiles and levels](#)
- [2.14 Wavelets](#)



Video Test Equipment

[▶ Video Test Home](#)

Product Portfolio

[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Related Products](#)
[Product Finder](#)

Resources For You

[Technical Papers](#)
[Manuals](#)
[Support Tools](#)
[Service](#)
[Software & Drivers](#)
[RGB Newsletter](#)
[Ask the Experts](#)

Section 3 Audio Compression

[3.1 The hearing mechanism](#)
[3.2 Subband coding](#)
[3.3 MPEG Layer 1](#)
[3.4 MPEG Layer 2](#)
[3.5 Transform coding](#)
[3.6 MPEG Layer 3](#)
[3.7 AC-3](#)

Section 4 Elementary Streams

[4.1 Video elementary stream syntax](#)
[4.2 Audio elementary streams](#)

Section 5 Packetized Elementary Streams (PES)

[5.1 PES packets](#)
[5.2 Time stamps](#)
[5.3 PTS/DTS](#)



Product Portfolio

[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Related Products](#)
[Product Finder](#)

Resources For You

[Technical Papers](#)
[Manuals](#)
[Support Tools](#)
[Service](#)
[Software & Drivers](#)
[RGB Newsletter](#)
[Ask the Experts](#)

What's New

None

Section 6 Program Streams

[6.1 Recording vs. transmission](#)
[6.2 Introduction to program streams](#)

Section 7 Transport Streams

[7.1 The job of a transport stream](#)
[7.2 Packets](#)
[7.3 Program Clock Reference \(PCR\)](#)
[7.4 Packet Identification \(PID\)](#)
[7.5 Program Specific Information \(PSI\)](#)

Section 8 Introduction to DVB/ATSC

[8.1 An overall view](#)
[8.2 Remultiplexing](#)
[8.3 Service Information \(SI\)](#)
[8.4 Error correction](#)
[8.5 Channel coding](#)
[8.6 Inner coding](#)
[8.7 Transmitting digits](#)



Video Test Equipment

[▶ Video Test Home](#)

Product Portfolio

[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Related Products](#)
[Product Finder](#)

Resources For You

[Technical Papers](#)
[Manuals](#)
[Support Tools](#)
[Service](#)
[Software & Drivers](#)
[RGB Newsletter](#)
[Ask the Experts](#)

What's New

[News](#)
[Events](#)

Section 9 MPEG Testing

[9.1 Testing requirements](#)
[9.2 Analyzing a Transport Stream](#)
[9.3 Hierarchic view](#)
[9.4 Interpreted view](#)
[9.5 Syntax and CRC analysis](#)
[9.6 Filtering](#)
[9.7 Timing Analysis](#)
[9.8 Elementary stream testing](#)
[9.9 Sarnoff compliant bit streams](#)
[9.10 Elementary stream analysis](#)
[9.11 Creating a transport stream](#)
[9.12 Jitter generation](#)
[9.13 DVB tests](#)

Glossary



Tektronix Measurement products are
manufactured in ISO registered facilities.



Video Test Equipment

[▶ Video Test Home](#)

Product Portfolio

[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Related Products](#)
[Product Finder](#)

Resources For You

[Technical Papers](#)
[Manuals](#)
[Support Tools](#)
[Service](#)
[Software & Drivers](#)
[RGB Newsletter](#)
[Ask the Experts](#)

What's New

[News](#)
[Events](#)

[9.13 DVB tests](#)

Glossary



Tektronix Measurement products are
manufactured in ISO registered facilities.

10/97 25W-11418-0A



[Email this
page](#)



[URL for
Bookmarks](#)



[Printable
page](#)

Tektronix Site

[Home](#) | [Products](#) | [Support](#) | [Sitemap](#) | [Search](#)

[© Copyright Tektronix, Inc.](#) | [Contact Us](#) | [Privacy
Statement](#)

← → ↻ <https://web.archive.org/web/20021225003145/http://www.tektronix.com/Measurement/cgi-bin/framed.pl> 133% ☆

http://www.tektronix.com/Measurement/cgi-bin/framed.pl?Document=/Measurement/App_Notes/mpegfund/&FrameSet=telev Go AUG DEC APR
19 captures 2001 25 2002 2004 About
7 Aug 2001 – 15 Jul 2014

Tektronix
Enabling Innovation

home products support buy contact us investor relations careers international my profile

Video Test Equipment ▶ [Video Test Home](#)

Search **GO**

Product Portfolio
[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Handheld Solutions](#)
[Request a Catalog](#)
[Product Finder](#)

Resources
[Technical Papers](#)
[Manuals](#)
[Ask the Experts](#)
[T&M; Fundamentals](#)
[Software & Drivers](#)
[Service & Support](#)
[Email Newsletters](#)

What's New
[Promotions](#)



A Guide to MPEG Fundamentals and Protocol Analysis

Step 1 of 2

MPEG is one of the most popular audio/video compression techniques because it is not just a single standard. Instead it is a range of standards suitable for different applications but based on similar principles. MPEG is an acronym for the Moving Picture Experts Group which was set up by the ISO (International Standards Organization) to work on compression.

To download "A Guide to MPEG Fundamentals and Protocol Analysis," please complete this form.

Tektronix
Enabling Innovation

[home](#) [products](#) [support](#) [buy](#) [contact us](#) [investor relations](#) [careers](#) [international](#) [my profile](#)

Video Test Equipment [▶ Video Test Home](#)

Search

GO

Product Portfolio
[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Handheld Solutions](#)
[Request a Catalog](#)
[Product Finder](#)

Resources
[Technical Papers](#)
[Manuals](#)
[Ask the Experts](#)
[T&M; Fundamentals](#)
[Software & Drivers](#)
[Service & Support](#)

*** required fields**

Email Address*

What standards do you test? (Check all that apply).

Video Format-

☐ -SDI
☐ -PAL
☐ -HD

Transmission-

☐ -COFDM
☐ -8VSB
☐ -QPSK

☐ -QAM
☐ -IP
☐ -ATM

Tektronix
Enabling Innovation

[home](#) [products](#) [support](#) [buy](#) [contact us](#) [investor relations](#) [careers](#) [international](#) [my profile](#)

Video Test Equipment [▶ Video Test Home](#)

Search
 GO

Product Portfolio
[Video Measurement](#)
[Signal Generators](#)
[Signal Monitors](#)
[MPEG Test](#)
[Picture Quality](#)
[RF Modulation](#)
[Audio Test](#)
[Handheld Solutions](#)
[Request a Catalog](#)
[Product Finder](#)

Resources
[Technical Papers](#)
[Manuals](#)
[Ask the Experts](#)
[T&M; Fundamentals](#)
[Software & Drivers](#)
[Service & Support](#)
[Email Newsletters](#)

What's New
[Promotions](#)
..

Protocol-
☐ -MPEG2
☐ -MPEG4
Other-

Please select the statement that best describes your situation (one only).*
☐ I plan to purchase within the next 6 months.
☐ I plan to purchase within the next 6-12 months.
☐ I have no timeline or plans for purchasing at this time.
☐ **Please have an account manager contact me.**

New Complexity Scalable MPEG Encoding Techniques for Mobile Applications

Stephan Mientens

*Philips Research Laboratories, Prof. Holstlaan 4, NL-5656 AA Eindhoven, The Netherlands
Email: stephan.mientens@philips.com*

Peter H. N. de With

*LogicaCMG Eindhoven, Eindhoven University of Technology, P.O. Box 7089, Luchthavenweg 57,
NL-5600 MB Eindhoven, The Netherlands
Email: p.h.n.de.with@tue.nl*

Christian Hentschel

*Cottbus University of Technology, Universitätsplatz 3-4, D-03044 Cottbus, Germany
Email: christian.hentschel@tu-cottbus.de*

Received 10 December 2002; Revised 7 July 2003

Complexity scalability offers the advantage of one-time design of video applications for a large product family, including mobile devices, without the need of redesigning the applications on the algorithmic level to meet the requirements of the different products. In this paper, we present complexity scalable MPEG encoding having core modules with modifications for scalability. The interdependencies of the scalable modules and the system performance are evaluated. Experimental results show scalability giving a smooth change in complexity and corresponding video quality. Scalability is basically achieved by varying the number of computed DCT coefficients and the number of evaluated motion vectors, but other modules are designed such they scale with the previous parameters. In the experiments using the “Stefan” sequence, the elapsed execution time of the scalable encoder, reflecting the computational complexity, can be gradually reduced to roughly 50% of its original execution time. The video quality scales between 20 dB and 48 dB PSNR with unity quantizer setting, and between 21,5 dB and 38,5 dB PSNR for different sequences targeting 1500 kbps. The implemented encoder and the scalability techniques can be successfully applied in mobile systems based on MPEG video compression.

Keywords and phrases: MPEG encoding, scalable algorithms, resource scalability.

1. INTRODUCTION

Nowadays, digital video applications based on MPEG video compression (e.g., Internet-based video conferencing) are popular and can be found in a plurality of consumer products. While in the past, mainly TV and PC systems were used, having sufficient computing resources available to execute the video applications, video is increasingly integrated into devices such as portable TV and mobile consumer terminals (see Figure 1).

Video applications that run on these products are heavily constrained in many aspects due to their limited resources as compared to high-end computer systems or high-end consumer devices. For example, real-time execution has to be assured while having limited computing power and memory for intermediate results. Different video resolutions have to be handled due to the variable displaying of video

frame sizes. The available memory access or transmission bandwidth is limited as the operating time is shorter for computation-intensive applications. Finally the product success on the market highly depends on the product cost. Due to these restrictions, video applications are mainly redesigned for each product, resulting in higher production cost and longer time-to-market.

In this paper, it is our objective to design a scalable MPEG encoding system, featuring scalable video quality and a corresponding scalable resource usage [1]. Such a system enables advanced video encoding applications on a plurality of low-cost or mobile consumer terminals, having limited resources (available memory, computing power, stand-by time, etc.) as compared to high-end computer systems or high-end consumer devices. Note that the advantage of scalable systems is that they are designed once for a whole product family instead of a single product, thus they have a faster



FIGURE 1: Multimedia applications shown on different devices sharing the available resources.

time-to-market. State-of-the-art MPEG algorithms do not provide scalability, thereby hampering, for example, low-cost solutions for portable devices and varying coding applications in multitasking environments.

This paper is organized as follows. Section 2 gives a brief overview of the conventional MPEG encoder architecture. Section 3 gives an overview of the potential scalability of computational complexity in MPEG core functions. Section 4 presents a scalable discrete cosine transformation (DCT) and motion estimation (ME), which are the core functions of MPEG coding systems. Part of this work was presented earlier. A special section between DCT and ME is devoted to content-adaptive processing, which is of benefit for both core functions. The enhancements on the system level are presented in Section 5. The integration of several individual scalable functions into a full scalable coder has given a new framework for experiments. Section 6 concludes the paper.

2. CONVENTIONAL MPEG ARCHITECTURE

The MPEG coding standard is used to compress a video sequence by exploiting the spatial and temporal correlations of the sequence as briefly described below.

Spatial correlation is found when looking into individual video frames (pictures) and considering areas of similar data structures (color, texture). The DCT is used to decorrelate spatial information by converting picture blocks to the transform domain. The result of the DCT is a block of transform coefficients, which are related to the frequencies contained in the input picture block. The patterns shown in Figure 2 are the representation of the frequencies, and each picture block is a linear combination of these basis patterns. Since high frequencies (at the bottom right of the figure) commonly have lower amplitudes than other frequencies and are less perceptible in pictures, they can be removed by quantizing the DCT coefficients.

Temporal correlation is found between successive frames of a video sequence when considering that the objects and background are on similar positions. For data compression purpose, the correlation is removed by predicting the contents and coding the frame differences instead of complete

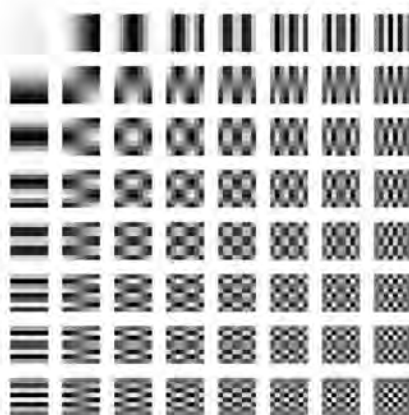


FIGURE 2: DCT block of basis patterns.

frames, thereby saving bandwidth and/or storage space. Motion in video sequences introduced by camera movements or moving objects result in high spatial frequencies occurring in the frame difference signal. A high compression rate is achieved by predicting picture contents using ME and motion compensation (MC) techniques.

For each frame, the above-mentioned correlations are exploited differently. Three different types of frames are defined in the MPEG coding standard, namely, I-, P-, and B-frames. I-frames are coded as completely independent frames, thus only spatial correlations are exploited. For P- and B-frames, temporal correlations are exploited, where P-frames use one temporal reference, namely, the past reference frame. B-frames use both the past and the upcoming reference frames, where I-frames and P-frames serve as reference frames. After MC, the frame difference signals are coded by DCT coding.

A conventional MPEG architecture is depicted in Figure 3. Since B-frames refer to future reference frames, they cannot be encoder/decoder before this reference frame is received by the coder (encoder or decoder). Therefore, the video frames are processed in a reordered way, for example, "IPBB" (transmit order) instead of "IBBP" (display order).

An algorithm-specific optimization that can be applied on any hardware architecture is to scale down the number of DCT coefficients that are computed. A new technique, considering the baseline DCT algorithm and a corresponding architecture for finding a specific computation order of the coefficients, is described in Section 4.1. The computation order maximizes the number of computed coefficients for a given limited amount of computation resources.

Another approach for scalable DCT computation predicts at several stages during the computation whether a group of DCT coefficients are zero after quantization and their computation can be stopped or not [3].

Inverse discrete cosine transformation

The IDCT transforms the DCT coefficients back to the spatial domain in order to reconstruct the reference frames for the (ME) and (MC) process. The previous discussion on scalability options for the DCT also applies to the IDCT. However, it should be noted that a scaled IDCT should have the same result as a perfect IDCT in order to be compatible with the MPEG standard. Otherwise, the decoder (at the receiver side) should ensure that it uses the same scaled IDCT as in the encoder in order to avoid error drift in the decoded video sequence.

Previous work on scalability of the IDCT at the receiver side exists [4, 5], where a simple subset of the received DCT coefficients is decoded. This has not been elaborated because in this paper, we concentrate on the encoder side.

Quantization

The quantization reduces the accuracy of the DCT coefficients and is therefore able to remove or weight frequencies of lower importance for achieving a higher compression ratio. Compared to the DCT where data dependencies during the computation of 64 coefficients are exploited, the quantization processes single coefficients where intermediate results cannot be reused for the computation of other coefficients. Nevertheless, computing the quantization involves rounding that can be simplified or left out for scaling up the processing speed. This possibility has not been worked out further.

Instead, we exploit scalability for the quantization based on the scaled DCT by preselecting coefficients for the computation such that coefficients that are not computed by the DCT are not further processed.

Inverse quantization

The inverse quantization restores the quantized coefficient values to the regular amplitude range prior to computing the IDCT. Like the IDCT, the inverse quantization requires sufficient accuracy to be compatible with the MPEG standard. Otherwise, the decoder at the receiver should ensure that it avoids error drift.

Motion estimation

The ME computes motion vector (MV) fields to indicate block displacements in a video sequence. A picture block

(macroblock) is then coded with reference to a block in a previously decoded frame (the prediction) and the difference to this prediction. The ME contains several scalability options. In principle, any good state-of-the-art fast ME algorithm offers an important step in creating a scaled algorithm. Compared to full search, the computing complexity is much lower (significantly less MV candidates are evaluated) while accepting some loss in the frame prediction quality. Taking the fast ME algorithms as references, a further increase of the processing speed is obtained by simplifying the applied set of motion vectors (MVs).

Besides reducing the number of vector candidates, the displacement error measurement (usually the sum of absolute pixel differences (SAD)) can be simplified (thus increase computation speed) by reducing the number of pixel values (e.g., via subsampling) that are used to compute the SAD. Furthermore, the accuracy of the SAD computation can be reduced to be able to execute more than one operation in parallel. As described for the DCT, this technique is suitable for a few hardware architectures only.

Up to this point, we have assumed that ME is performed for each macroblock. However, the number of processed macroblocks can be reduced also, similar to the pixel count for the SAD computation. MVs for omitted macroblocks are then approximated from neighboring macroblocks. This technique can be used for concentrating the computing effort on areas in a frame, where the block contents lead to a better estimation of the motion when spending more computing power [6].

A new technique to perform the ME in three stages by exploiting the opportunities of high-quality frame-by-frame ME is presented in Section 4.3. In this technique, we used several of the above-mentioned options and we deviate from the conventional MPEG processing order.

Motion compensation

The MC uses the MV fields from the ME and generates the frame prediction. The difference between this prediction and the original input frame is then forwarded to the DCT. Like the IDCT and the inverse quantization, the MC requires sufficient accuracy for satisfying the MPEG standard. Otherwise, the decoder (at the receiver) should ensure using the same scaled MC as in the encoder to avoid error drift.

Variable-length coding (VLC)

The VLC generates the coded video stream as defined in the MPEG standard. Optimization of the output can be made here, like ensuring a predefined bit rate. The computational effort is scalable with the number of nonzero coefficients that remain after quantization.

4. SCALABLE FUNCTIONS FOR MPEG ENCODING

Computationally expensive corner stones of an MPEG encoder are the DCT and the ME. Both are addressed in the scalable form in Section 4.1 on the scalable DCT [7] and in Section 4.3 on the scalable ME [8], respectively. Additionally,

Section 4.2 presents a scalable block classification algorithm, which is designed to support and integrate the scalable DCT and ME on the system level (see Section 5).

4.1. Discrete Cosine Transformation

4.1.1. Basics

The DCT transforms the luminance and chrominance values of small square blocks of an image to the transform domain. Afterwards, all coefficients are quantized and coded. For a given $N \times N$ image block represented as a two-dimensional (2D) data matrix $\{X[i, j]\}$, where $i, j = 0, 1, \dots, N-1$, the 2D DCT matrix of the coefficients $\{Y[m, n]\}$ with $m, n = 0, 1, \dots, N-1$ is computed by

$$Y[m, n] = \frac{4}{N^2} * u(m) * u(n) * \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X[i, j] * \cos \frac{(2i+1)m * \pi}{2N} * \cos \frac{(2j+1)n * \pi}{2N}, \quad (1)$$

where $u(i) = 1/\sqrt{2}$ if $i = 0$ and $u(i) = 1$ elsewhere. Equation (1) can be simplified by ignoring the constant factors for convenience and defining a square cosine matrix K by

$$K_N[p, q] = \cos \frac{(2p+1)q * \pi}{2N} \quad (2)$$

so that (1) can be rewritten as

$$Y = K_N * X * K_N^T. \quad (3)$$

Equation (3) shows that the 2D DCT as specified by (1) is based on two orthogonal 1D DCTs, where $K_N * X$ transforms the columns of the image block X , and $X * K_N^T$ transforms the rows. Since the computation of two 1D DCTs is less expensive than one 2D DCT, state-of-the-art DCT algorithms normally refer to (3) and concentrate on optimizing a 1D DCT.

4.1.2. Scalability

Our proposed scalable DCT is a novel technique for finding a specific computation order of the DCT coefficients. The results depend on the applied (fast) DCT algorithm. In our approach, the DCT algorithm is modified by eliminating several computations and thus coefficients, thereby enabling complexity scalability for the used algorithm. Consequently, the output of the algorithm will have less quality, but the processing effort of the algorithm is reduced, leading to a higher computing speed. The key issue is to identify the computation steps that can be omitted to maximize the number of coefficients for the best possible video quality.

Since fast DCT algorithms process video data in different ways, the algorithm used for a certain scalable application should be analyzed closely as follows. Prior to each computation step, a list of remaining DCT coefficients is sorted

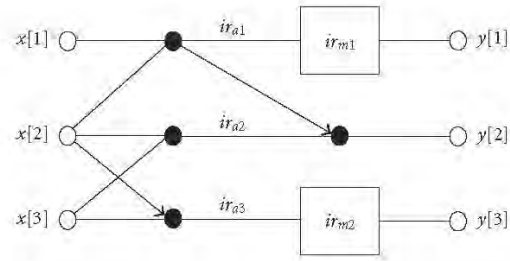


FIGURE 4: Exemplary butterfly structure for the computation of outputs $y[\cdot]$ based on inputs $x[\cdot]$. The data flow of DCT algorithms can be visualized using such butterfly diagrams.

such that in the next step, the coefficient is computed having the lowest computational cost. More formally, the sorted list $L = \{l_1, l_2, \dots, l_{N^2}\}$ of coefficients l taken from an $N \times N$ DCT satisfies the condition

$$C(l_i) = \min_{k \geq i} C(l_k), \quad \forall l_i \in L \quad (4)$$

where $C(l_k)$ is a cost function providing the remaining number of operations required for the coefficient l_k given the fact that the coefficients l_n , $n < k$, already have been computed. The underlying idea is that some results of previously performed computations can be shared. Thus (4) defines a minimum computational effort needed to obtain the next coefficient.

We give a short example of how the computation order L is obtained. In Figure 4, a computation with six operation nodes is shown, where three nodes are intermediate results (ir_{a1} , ir_{a2} , and ir_{a3}). The complexity of the operations that are involved for a node can be defined such that they represent the characteristics (like CPU usage or memory access costs) of the target architecture. For this example, we assume that the nodes depicted with filled circles ($\bullet$) require one operation and nodes that are depicted with squares ($\square$) require three operations. Then, the outputs (coefficients) $y[1]$, $y[2]$, and $y[3]$ require 4, 3, and 4 operations, respectively. In this case, the first coefficient in list L is $l_1 = y[2]$ because it requires the least number of operations. Considering that, with $y[2]$, the shared node ir_{a1} has been computed and its intermediate result is available, the remaining coefficients $y[1]$ and $y[3]$ require 3 and 4 operations, respectively. Therefore, $l_2 = y[1]$ and $l_3 = y[3]$, leading to a computation order $L = \{y[2], y[1], y[3]\}$.

The computation order L can be perceptually optimized if the subsequent quantization step is considered. The quantizer weighting function emphasizes the use of low-frequency coefficients in the upper-left corner of the matrix. Therefore, the cost function $C(l_k)$ can be combined with a priority function to prefer those coefficients.

Note that the computation order L is determined by the algorithm and the optional applied priority function, and it can be found in advance. For this reason, no computational

	0	1	2	3	4	5	6	7
0	1	33	9	41	5	44	14	36
1	17	49	21	57	29	63	31	55
2	10	37	3	42	11	39	7	48
3	25	61	26	53	18	51	24	60
4	6	45	15	34	2	35	16	46
5	28	59	23	52	19	54	27	62
6	12	47	8	40	13	43	4	38
7	20	56	32	64	30	58	22	50

FIGURE 5: Computation order of coefficients.

overhead is required for actually computing the scaled DCT. It is possible, though, to apply different precomputed DCTs to different blocks employing block classification that indicates which precomputed DCT should perform best with a classified block (see Section 5.3).

4.1.3. Experiments

For experiments, the fast 2D algorithm given by Cho and Lee [9], in combination with the Arai-Agui-Nakajima (AAN) 1D algorithm [10], has been used, and this algorithm combination is extended in the following with computational complexity scalability. Both algorithms were adopted because their combination provides a highly efficient DCT computation (104 multiplications and 466 additions). The results of this experiment presented below are discussed with the assumption that an addition is equal to one operation and a multiplication is equal to three operations (in powerful cores, additions and multiplications have equal weight).

The scalability-optimized computation order in this experiment is shown in Figure 5, where the matrix has been shaded with different gray levels to mark the first and the second half of the coefficients in the sorted list. It can be seen that in this case, the computation order clearly favors horizontal or vertical edges (depending on whether the matrix is transposed or not).

Figure 6 shows the scalability of our DCT computation technique using the scalability-optimized computation order, and the zigzag order as reference computation order. In Figure 6a, it can be seen that the number of coefficients that are computed with the scalability-optimized computation order is higher at any computation limit than the zigzag order. Figure 6b shows the resulting peak signal-to-noise ratio (PSNR) of the first frame from the “Voit” sequence using both computation orders, where no quantization step is performed. A 1–5 dB improvement in PSNR can be noticed, depending on the amount of available operations.

Finally, Figure 7 shows two picture pairs (based on zigzag and scalability-optimized orders preferring horizontal details) sampled from the “Renata” sequence during different stages of the computation (representing low-cost and medium-cost applications). Perceptive evaluations of our ex-

periments have revealed that the quality improvement of our technique is the largest between 200 and 600 operations per block. In this area, the amount of coefficients is still relatively small so that the benefit of having much more coefficients computed than in a zigzag order is fully exploited. Although the zigzag order yields perceptually important coefficients from the beginning, the computed number is simply too low to show relevant details (e.g., see the background calendar in the figure).

4.2. Scalable classification of picture blocks

4.2.1. Basics

The conventional MPEG encoding system processes each image block in the same content-independent way. However, content-dependent processing can be used to optimize the coding process and output quality, as indicated below.

- (i) Block classification is used for quantization to distinguish between flat, textured, and mixed blocks [11] and then apply different quantization factors for these blocks for optimizing the picture quality at given bit rate limitations. For example, quantization errors in textured blocks have a small impact on the perceived image quality. Blocks containing both flat and textured parts (mixed blocks) are usually blocks that contain an edge, where the disturbing ringing effect gets worse with high quantization factors.
- (ii) The ME (see Section 4.3) can take the advantage of classifying blocks to indicate whether a block has a structured content or not. The drawback of conventional ME algorithms that do not take the advantage of block classification is that they spend many computations on computing MVs for, for example, relatively flat blocks. Unfortunately, despite the effort, such ME processes yield MVs of poor quality. Employing block classification, computations can be concentrated on blocks that may lead to accurate MVs [12].

Of course, in order to be useful, the costs to perform block classification should be less than the saved computations. Given the above considerations, in the following, we will adopt content-dependent adaptivity for coding and motion processing. The next section explains the content adaptivity in more detail.

4.2.2. Scalability

We perform a simple block classification based on detecting horizontal and vertical transitions (edges) for two reasons.

- (i) From the scalable DCT, computation orders are available that prefer coefficients representing horizontal or vertical edges. In combination with a classification, the computation order that fits best for the block content can be chosen.
- (ii) The ME can be provided with the information whether it is more likely to find a good MV in up-down or left-right search directions. Since ME will find equally

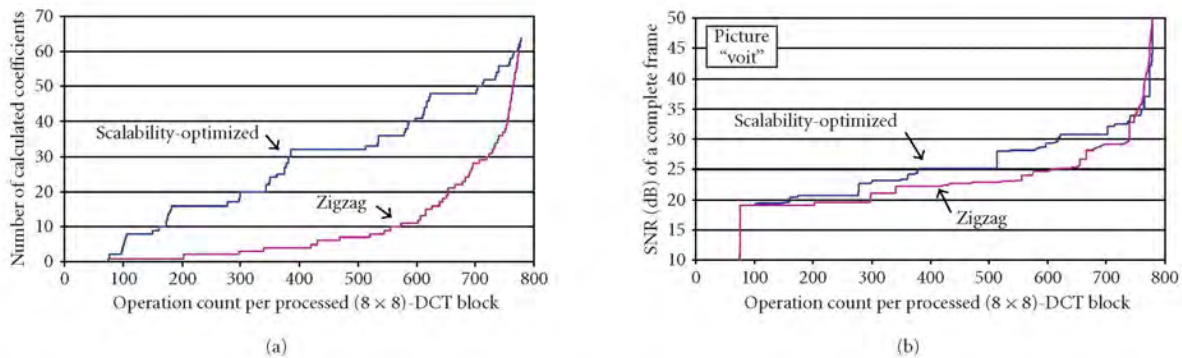


FIGURE 6: Comparison of the scalability-optimized computation order with the zigzag order. At limited computation resources, more DCT coefficients are computed (a) and a higher PSNR is gained (b) with the scalability-optimized order than with the zigzag order.

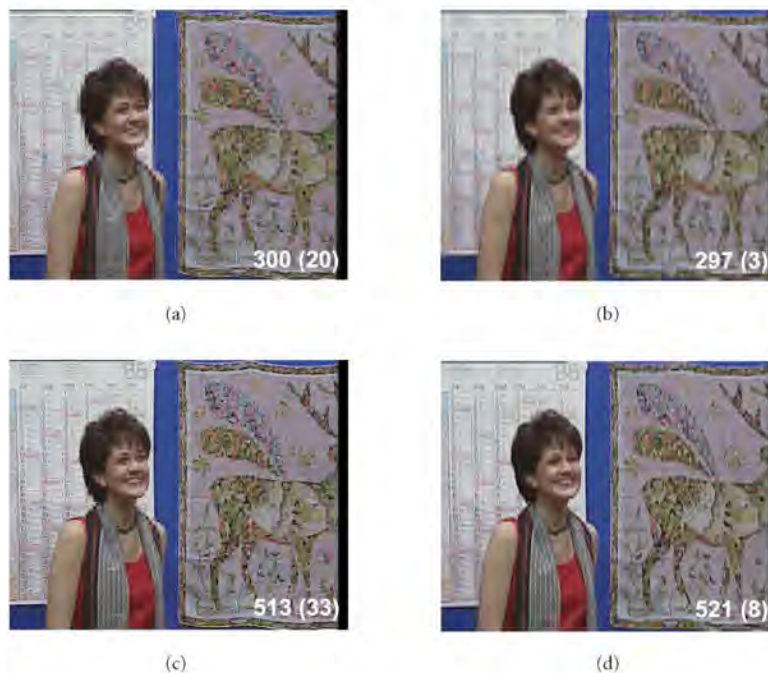


FIGURE 7: A video frame from the "Renata" sequence coded employing the scalability-optimized order (a) and (c), and the zigzag order (b) and (d). Index $m(n)$ means m operations are performed for n coefficients. The scalability-optimized computation order results in an improved quality (compare sharpness and readability).

good MVs for every position *along* such an edge (where a displacement in this direction does not introduce large displacement errors), searching for MVs *across* this edge will rapidly reduce the displacement error and thus lead to an appropriate MV. Horizontal and vertical edges can be detected by significant changes of pixel values in vertical and horizontal directions, respectively.

The edge detecting algorithm we use is in principle based on continuously summing up pixel differences along rows or columns and counting how often the sum exceeds a certain threshold. Let p_i , with $i = 0, 1, \dots, 15$, be the pixel values in a row or column of a macroblock (size 16×16). We then define a range where pixel divergence (d_i) is considered as noise if $|d_i|$ is below a threshold t . The pixel divergence is defined by Table 1.

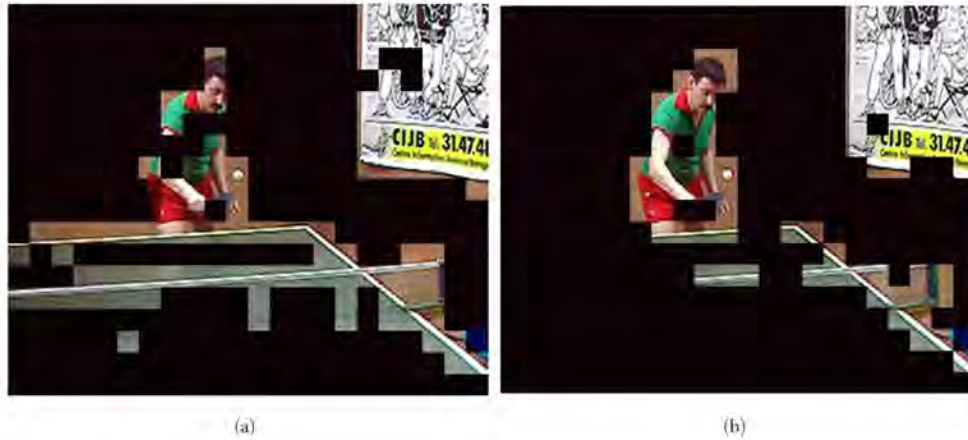


FIGURE 8: Visualization of block classification using a picture of the “table tennis” sequence. The left (right) picture shows blocks where horizontal (vertical) edges are detected. Blocks that are visible in both pictures belong to the class “diagonal/structured,” while blocks that are blanked out in both pictures are considered as “flat.”

TABLE 1: Definition of pixel divergence, where the divergence is considered as noise if it is below a certain threshold.

Condition	Pixel divergence $\bar{d}_i$
$i = 0$	0
$(i = 1, \dots, 15) \wedge (d_{i-1} \leq t)$	$d_{i-1} + (p_i - p_{i-1})$
$(i = 1, \dots, 15) \wedge (d_{i-1} > t)$	$d_{i-1} + (p_i - p_{i-1}) - \text{sgn}(d_{i-1}) * t$

The area preceding the edge yields a level in the interval $[-t; +t]$. The middle of this interval is at $d = 0$, which is modified by adding $\pm t$ in the case that $|d|$ exceeds the interval around zero (start of the edge). This mechanism will follow the edges and prevent noise from being counted as edges. The counter c as defined below indicates how often the actual interval was exceeded:

$$c = \sum_{i=1}^{15} \begin{cases} 0 & \text{if } |d_i| \leq t, \\ 1 & \text{if } |d_i| > t. \end{cases} \quad (5)$$

The occurrence of an edge is defined by the resulting value of c from (5).

This edge detecting algorithm is scalable by selecting the threshold t , the number of rows and columns that are considered for the classification, and a typical value for c . Experimental evidence has shown that in spite of the complexity scalability of this classification algorithm, the evaluation of a single row or column in the middle of a picture block was found sufficient for a rather good classification.

4.2.3. Experiments

Figure 8 shows the result of an example to classify image blocks of size 16×16 pixels (macroblock size). For this ex-

periment, a threshold of $t = 25$ was used. We considered a block to be classified as a “horizontal edge” if $c \geq 2$ holds for the central column computation and as a “vertical edge” if $c \geq 2$ holds for the row computation. Obviously, we can derive two extra classes: “flat” (for all blocks that do not belong to the CLASS “horizontal edge” NOR the class “vertical edge”) and diagonal/structured (for blocks that belong to both classes horizontal edge and vertical edge).

The visual results of Figure 8 are just an example of a more elaborate set of sequences with which experiments were conducted. The results showed clearly that the algorithm is sufficiently capable of classifying the blocks for further content-adaptive processing.

4.3. Motion estimation

4.3.1. Basics

The ME process in MPEG systems divides each frame into rectangular macroblocks (16×16 pixels each) and computes MVs per block. An MV signifies the displacement of the block (in the x - y pixel plane) with respect to a reference image. For each block, a number of candidate MVs are examined. For each candidate, the block evaluated in the current image is compared with the corresponding block fetched from the reference image displaced by the MV. After testing all candidates, the one with the best match is selected. This match is done on basis of the SAD between the current block and the displaced block. The collection of MVs for a frame forms an MV field.

State-of-the-art ME algorithms [13, 14, 15] normally concentrate on reducing the number of vector candidates for a single-sided ME between two frames, independent of the frame distance. The problem of these algorithms is that a higher frame distance hampers accurate ME.

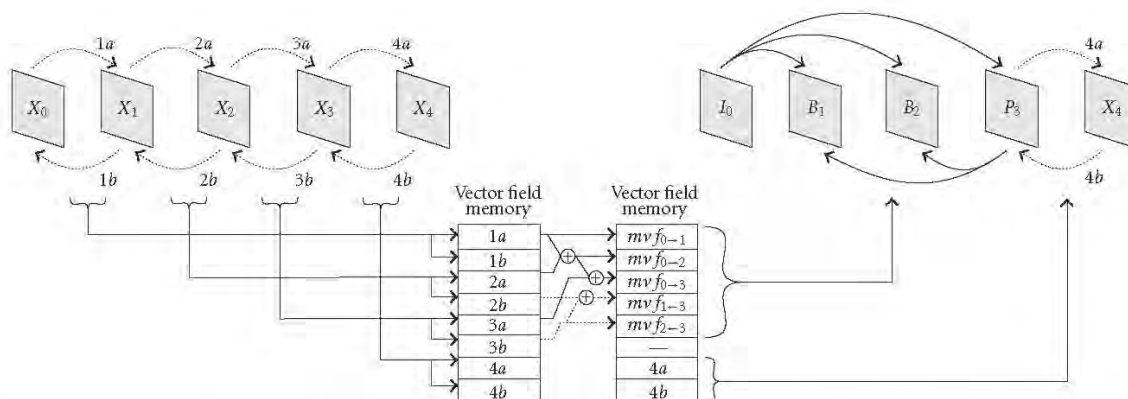


FIGURE 9: An overview of the new scalable ME process. Vector fields are computed for successive frames (left) and stored in memory. After defining the GOP structure, an approximation is computed (middle) for the vector fields needed for MPEG coding (right). Note that for this example it is assumed that the approximations are performed after the exemplary GOP structure is defined (which enables dynamic GOP structures), therefore the vector field (1b) is computed but not used afterwards. With predefined GOP structures, the computation of (1b) is not necessary.

4.3.2. Scalability

The scalable ME is designed such that it takes the advantage of the intrinsically high prediction quality of ME between successive frames (smallest temporal distance), and thereby works not only for the typical (predetermined and fixed) MPEG GOP structures, but also for more general cases. This feature enables on-the-fly selection of GOP structures depending on the video content (e.g., detected scene changes, significant changes of motion, etc.). Furthermore, we introduce a new technique for generating MV fields from other vector fields by *multitemporal* approximation (not to be confused with other forms of multitemporal ME as found in H.264). These new techniques give more flexibility for a scalable MPEG encoding process.

The estimation process is split up into three stages as follows.

- Stage 1 Prior to defining a GOP structure, we perform a simple recursive motion estimation (RME) [16] for every received frame to compute the forward and backward MV field between the received frame and its predecessor (see the left-hand side of Figure 9). The computation of MV fields can be omitted for reducing computational effort and memory.
- Stage 2 After defining a GOP structure, all the vector fields required for MPEG encoding are generated through multitemporal approximations by summing up vector fields from the previous stage. Examples are given in the middle of Figure 9, for example, vector field $(mvf_{0-3}) = (1a) + (2a) + (3a)$. Assume that the vector field (2a) has not been computed in Stage 1 (due to a chosen scalability setting), one possibility to approximate (mvf_{0-3}) is $(mvf_{0-3}) = 2 * (1a) + (3a)$.
- Stage 3 For final MPEG ME in the encoder, the computed approximated vector fields from the previous stage are

used as an input. Beforehand, an optional refinement of the approximations can be performed with a second iteration of simple RME.

We have employed simple RME as a basis for introducing scalability because it offers a good quality for time-consecutive frames at low computing complexity.

The presented three-stage ME algorithm differs from known multistep ME algorithms like in [17], where initially estimated MPEG vector fields are processed for a second time. Firstly, we do not have to deal with an increasing temporal distance when deriving MV fields in Stage 1. Secondly, we process the vector fields in a display order having the advantage of frame-by-frame ME, and thirdly, our algorithm provides scalability. The possibility of scaling vector fields, which is part of our multitemporal predictions, is mentioned in [17] but not further exploited. Our algorithm makes explicit use of this feature, which is a fourth difference. In the sequel, we explain important system aspects of our algorithm.

Figure 10 shows the architecture of the three-stage ME algorithm embedded in an MPEG encoder. With this architecture, the initial ME process in Stage 1 results in a high-quality prediction because original frames without quantization errors are used. The computed MV fields can be used in Stage 2 to optimize the GOP structures. The optional refinement of the vector fields in Stage 3 is intended for high-quality applications to reach the quality of a conventional MPEG ME algorithm.

The main advantage of the proposed architecture is that it enables a broad scalability range of resource usage and achievable picture quality in the MPEG encoding process. Note that a bidirectional ME (usage of B-frames) can be realized at the same cost of a single-directional ME (usage of P-frames only) when properly scaling the computational

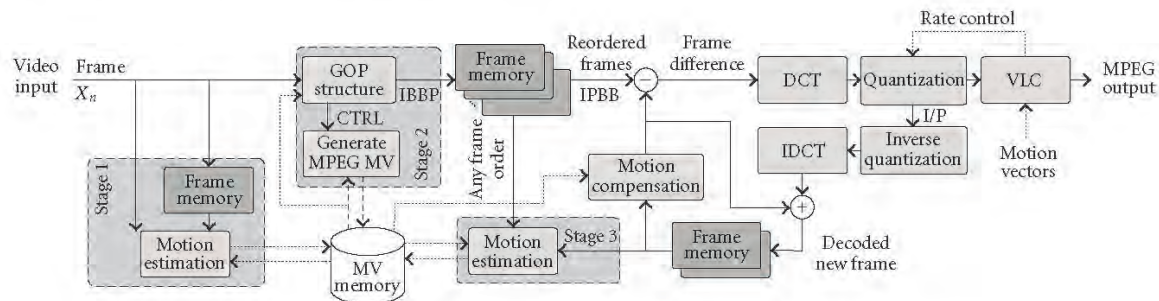
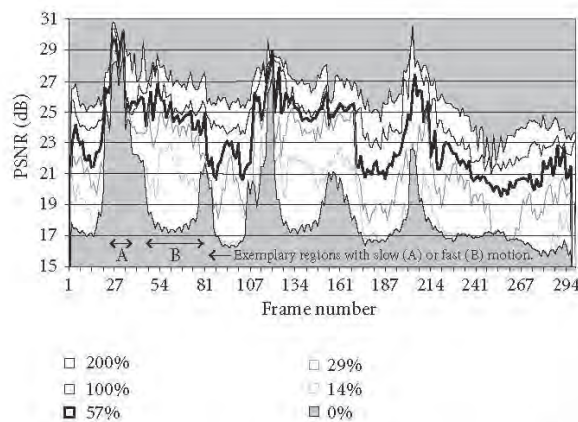


FIGURE 10: Architecture of an MPEG encoder with the new scalable three-stage motion estimation.

FIGURE 11: PSNR of motion-compensated B-frames of the "Stefan" sequence (tennis scene) at different computational efforts—P-frames are not shown for the sake of clarity ($N = 16$, $M = 4$). The percentage shows the different computational effort that results from omitting the computation of vector fields in Stage 1 or performing an additional refinement in Stage 3.

complexity, which makes it affordable for mobile devices that up till now rarely make use of B-frames. A further optimization is seen (but not worked out) in limiting the ME process of Stages 1 and 3 to significant parts of a vector field in order to further reduce the computational effort and memory.

4.3.3. Experiments

To demonstrate the flexibility and scalability of the three-stage ME technique, we conducted an initial experiment using the "Stefan" sequence (tennis scene). A GOP size of $N = 16$ and $M = 4$ (thus "IBBBP" structure) was used, combined with a simple pixel-based search. In this experiment, the scaling of the computational complexity is introduced by gradually increasing the vector field computations in Stage 1 and Stage 3. The results of this experiment are shown in Figure 11. The area in the figure with the white background shows the scalability of the quality range that results from downscaling the amount of computed MV fields. Each vector

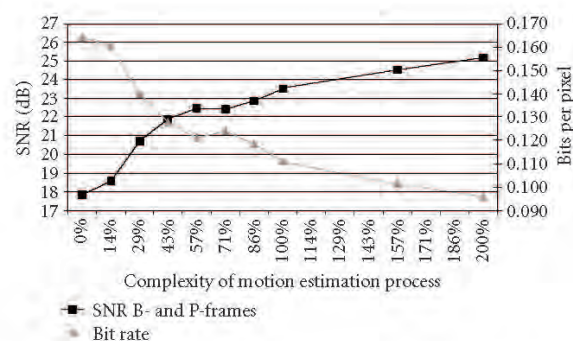


FIGURE 12: Average PSNR of motion-compensated P- and B-frames and the resulting bit rate of the encoded "Stefan" stream at different computational efforts. A lower average PSNR results in a higher differential signal that must be coded, which leads to a higher bit rate. The percentage shows the different computational effort that results from omitting the computation of vector fields in Stage 1 or performing an additional refinement in Stage 3.

field requires 14% of the effort compared to a 100% simple RME [16] based on four forward vector fields and three backward vector fields when going from one to the next reference frame. If all vector fields are computed and the refinement Stage 3 is performed, the computational effort is 200% (not optimized).

The average PSNR of the motion-compensated P- and B-frames (taken after MC and before computing the differential signal) of this experiment and the resulting bit rate of the encoded MPEG stream are shown in Figure 12. Note that for comparison purpose, no bit rate control is performed during encoding and therefore, the output quality of the MPEG streams for all complexity levels is equal. The quantization factors, $qscale$, we have used are 12 for I-frames and 8 for P- and B-frames. For a full quality comparison (200%), we consider a full-search block matching with a search window of 32×32 pixels. The new ME technique slightly outperforms this full search by 0.36 dB PSNR measured from the motion-compensated P- and B-frames of this experiment (25.16 dB instead of 24.80 dB). The bit rate of the complete MPEG

TABLE 2: Average luminance PSNR of the motion-compensated P- and B-frames for sequences “Stefan” (A), “Renata” (B), and “Teen” (C) with different ME algorithms. The second column shows the average number of SAD-based vector evaluations per MV (based on (A)).

Algorithm	Tests/MV	(A)	(B)	(C)
2D FS (32×32)	926.2	24.80	29.62	26.78
NTSS [14]	25.2	22.55	27.41	24.22
Diamond [15]	21.9	22.46	27.34	26.10
Simple RME [16]	16.0	21.46	27.08	23.89
Three-stage ME 200% (employing [16])	37.1	25.16	29.24	26.92
Three-stage ME 100% (employing [16])	20.1	23.52	27.45	24.74

sequence is 0.012 bits per pixel (bpp) lower when using the new technique (0.096 bpp instead of 0.108 bpp). When reducing the computational effort to 57% of a single-pass simple RME, an increase of the bit rate by 0.013 bpp compared to the 32×32 full search (FS) is observed.

Further comparisons are made with the scalable three-stage ME running at full and “normal” quality. Table 2 shows the average PSNR of the motion-compensated P- and B-frames for three different video sequences and ME algorithms with the same conditions as described above (same N , M , etc.). The first data column (tests per MV) shows the average number of vector tests that are performed per macroblock in the “Stefan” sequence to indicate the performance of the algorithms. Note that MV tests pointing outside the picture are not counted, which results in numbers that are lower than the nominal values (e.g., 926.2 instead of 1024 for 32×32 FS). The simple RME algorithm results in the lowest quality here because only three vector field computations out of $4 * (4 + 3) = 28$ can use temporal vector candidates as prediction. However, our new three-stage ME that uses this simple RME performs, comparable to FS, at 200% complexity, and at 100%, it is comparable to the other fast ME algorithms.

The results in Table 2 are based on the simple RME algorithm from [16]. A modified algorithm has been found later [18] that forms an improved replacement for the simple RME. This modified algorithm is based on the block classification as presented in Section 4.2. This algorithm was used for further experiments and is summarized as follows. Prior to estimating the motion between two frames, the macroblocks inside a frame are classified into areas having horizontal, vertical edges, or no edges. The classification is exploited to minimize the number of MV evaluations for each macroblock by, for example, concentrating vector evaluations across the detected edge. A novelty in the algorithm is a *distribution* of good MVs to other macroblocks, even already processed ones, which differs from other known recursive ME techniques that reuse MVs from previously processed blocks.

5. SYSTEM ENHANCEMENTS AND EXPERIMENTS

The key approach to optimize a system is to reuse and combine data that is generated by the system modules in order to control other modules. In the following, we present several

approaches, where data can be reused or generated at a low cost in a coding system for an optimization purpose.

5.1. Experimental environment

The scalable modules for the (I)DCT, (de)quantization, ME, and VLC are integrated into an MPEG encoder framework, where the scaling of the IDCT and the (de)quantization is effected from the scalable DCT (see Section 5.2). In order to visualize the obtained scalability of the computations, the scalable modules are executed at different parameter settings, leading to effectively varying the number of DCT coefficients and MV candidates evaluated. When evaluating the system complexity, the two different numbers have to be combined into a joint measure. In the following, the elapsed execution time of the encoder needed to code a video sequence is used as a basis for comparison. Although this time parameter highly depends on the underlying architecture and on the programming and operating system, it reflects the complexity of the system due to the high amount of operations involved.

The experiments were conducted on a Pentium-III Linux system running at 733 MHz. In order to be able to measure the execution time of single functions being part of the complete encoder execution, it was necessary to compile the C++ program of the encoder without compiler optimizations. Additionally, it should be noted that the experimental C++ code was not optimized for fast execution or usage of architecture-specific instructions (e.g., MMX). For these reasons, the encoder and its measured execution times cannot be compared with existing software-based MPEG encoders. However, we have ensured that the measured change in the execution time results from the scalability of the modules, as we did not change the programming style, code structures, or common coding parameters.

5.2. Effect of scalable DCT

The fact that a scaled DCT computes only a subset S of all possible DCT coefficients C can be used for the optimization of other modules. The subset S is known before the subsequent quantization, dequantization, VLC, and IDCT modules. Of course, coefficients that are not computed are set to zero and therefore they do not have to be processed further in any of these modules. Note that because the subset S is known in advance, no additional tests are performed to

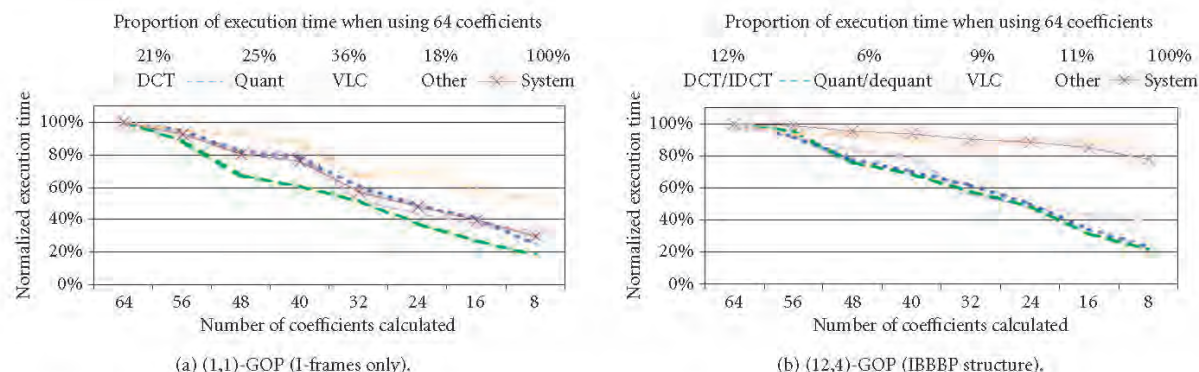


FIGURE 13: Complexity reduction of the encoder modules relative to the full DCT processing, with (1,1)-GOPs (a) and with (12,4)-GOPs (b). Note that in this case, 62% of the coding time is spent in (b) for ME and MC (not shown for convenience). For visualization of the complexity reduction, we normalize the execution time for each module to 100% for full processing.

detect zero coefficients. This saves computations as follows.

- (i) The quantization and dequantization require a fixed amount of operations per processed intra- or intercoefficient. Thus, each skipped coefficient $c \in C \setminus S$ saves 1/64 of the total complexity of the quantization and dequantization modules.
- (ii) The VLC processes the DCT coefficients in a zigzag or an alternate order and generates run-value pairs for coefficients that are unequal to zero. “Run” indicates the number of zero coefficients that are skipped before reaching a nonzero coefficient. The usage of a scaled DCT increases the probability that zero coefficients occur, for which no computations are spent.
- (iii) The IDCT can be simplified by knowing which coefficients are zero. It is obvious that, for example, each multiplication with a known factor of 0 and additions with a known addend of 0 can be skipped.

The execution time of the modules when coding the “Stefan” sequence and scaling the modules that process coefficients is visualized in Figure 13. The category “other” is used for functions that are not exclusively used by the scaled modules. Figure 13a shows the results of an experiment, where the sequence was coded with I-frames only. Similar results are observed in Figure 13b from another experiment, for which P- and B-frames are included. To remove the effect of quantization, the experiments were performed with $qscale = 1$. In this way, the figures show results that are less dependent on the coded video content.

The measured PSNR of the scalable encoder running at full quality is 46.5 dB for Figure 13a and 48.16 dB for Figure 13b. When the number of computed coefficients is gradually reduced from 64 to 8, the PSNR drops gradually to 21.4 dB Figure 13a, respectively, 21.81 dB in Figure 13b. In Figures 13a and 13b, the quality gradually reduces from “no noticeable differences” down to “severe blockiness.” In Figure 13b, the curve for the ME module is not shown for

convenient because the ME (in this experiment, we used diamond search ME [15]) is not affected from processing a different number of DCT coefficients.

5.3. Selective DCT computation based on block classification

The block classification introduced in Section 4.2 is used to enhance the output quality of the scaled DCT by using different computation orders for blocks in different classes. A simple experiment indicates the benefit in quality improvement. In the experiment, we computed the average values of DCT coefficients when coding the “table tennis” sequence with I-frames only. Each DCT block is taken after quantization with $qscale = 1$. Figure 14 shows the statistic for blocks that are classified as having a horizontal (left graph) or vertical (right graph) edge only. It can be seen that the classification leads to a frequency concentration in the DCT coefficient matrix in the first column, respectively, row.

We found that the DCT algorithm of Arai et al. [10] can be used best for blocks with horizontal or vertical edges, while background blocks have a better quality impression when using the algorithm by Cho and Lee [9]. The experiment made for Figure 15 shows the effect of the two algorithms on the table edges ([10] is better) and the background ([9] is better). In both cases, the computation orders designed for preferring horizontal edges are used. The computation limit was set to 256 operations, leading to 9 computed coefficients for [10] and 11 for [9], respectively. The coefficients that are computed are marked in the corresponding DCT matrix. It can be seen that [10] covers all main vertical frequencies, while [9] covers a mixture of high and low vertical and horizontal frequencies. The resulting overall PSNR are 26.58 dB and 24.32 dB, respectively.

Figure 16 shows the effect of adaptive DCT computation based on classification. Almost all of the background blocks were classified as flat blocks and therefore, ChoLee was chosen for these blocks. For convenient, both algorithms were set

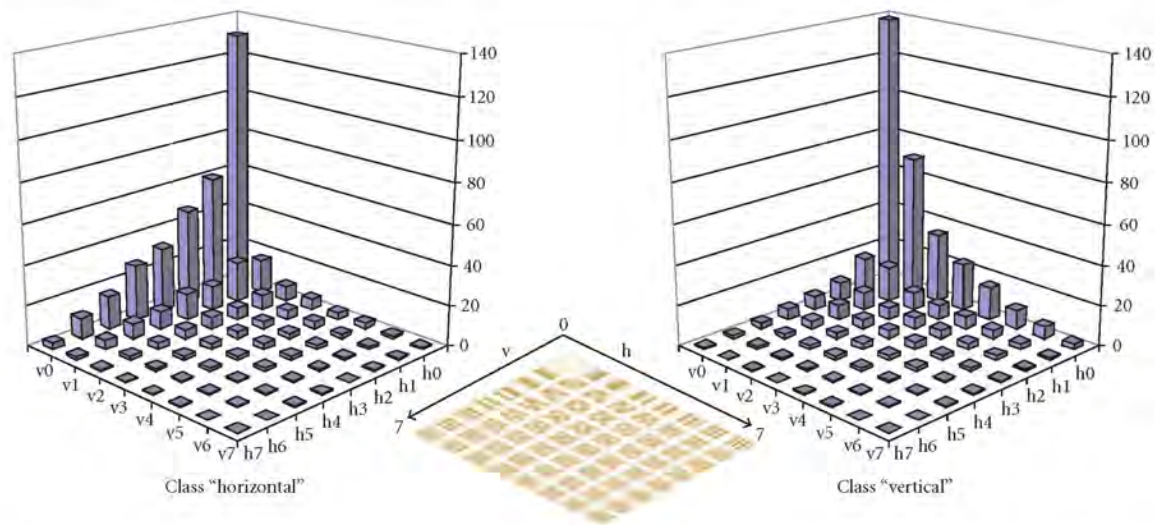


FIGURE 14: Statistics of the average absolute values of the DCT coefficients taken after quantization with $qscale = 1$. Here, the “table tennis” sequence was coded with I-frames only. The left (right) graph shows the statistic for blocks classified as having horizontal (vertical) edges.

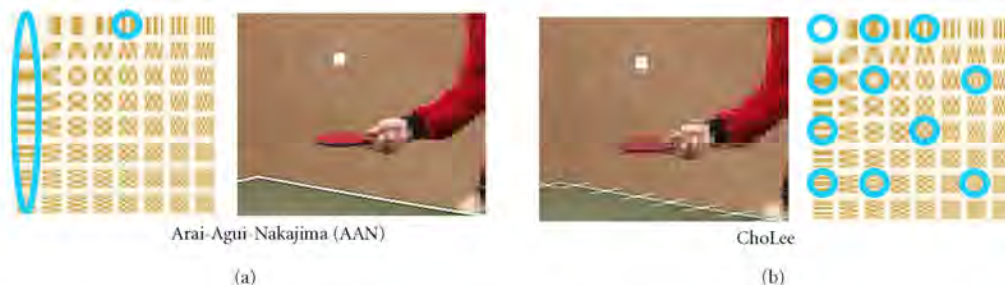


FIGURE 15: Example of scaled AAN-DCT (a) and ChoLee-DCT (b) at 256 operations. AAN fits better for horizontal edges, while ChoLee has better results for the background.

to compute 11 coefficients. Blocks with both detected horizontal and vertical edges are treated as blocks having horizontal edges only because an optimized computation order for such blocks is not yet defined. The resulting PSNR is 26.91 dB.

5.4. Dynamic interframe DCT coding

Besides intraframe coding, the DCT computation on frame differences (for interframe coding) occurs more often than intraframe coding ($N - 1$ times for (N, M) GOPs). For this reason, we look more closely to interframe DCT coding, where we discovered a special phenomenon from the scalable DCT. It was found that the DCT coded frame differences show temporal fluctuations in frequency content. The temporal fluctuation is caused by the motion in the video content combined with the special selection function of the coefficients computed in our scalable DCT. Due to the motion, the energy in the coefficients shifts over the selection pattern

so that the quality gradually increases over time. Figure 17 shows this effect from an experiment when coding the “Stefan” sequence with IPP frames (GOP structure (GOP size N , IP distance M) = (12, 1)) while limiting the computation to 32 coefficients. The camera movement in the shown sequence is panning to the right. It can be seen for example that the artifacts around text decrease over time.

The aforementioned phenomenon was mainly found in sequences containing not too much motion. The described effect leads to the idea of temporal data partitioning using a cyclical sequence of several scalable DCTs with different coefficient selection functions. The complete cycle would compute each coefficient at least once. Temporal data partitioning means that the computational complexity of the DCT computation is spread over time, thereby reducing the average complexity of the DCT computation (per block) at the expense of obtaining delayed quality attainment. Using this technique, picture blocks having a static content (blocks



FIGURE 16: Both DCT algorithms were used to code this frame. After block classification, the ChoLee-DCT was used to code blocks where no edges were detected and the AAN-DCT for blocks with detected edges.



FIGURE 17: Visualization of a phenomenon from the scalable DCT, leading to a gradual quality increase over time.

having zero motion like nonmoving background) and therefore having no temporal fluctuations in their frequency content will obtain the same result as a nonpartitioned DCT computation after full computation of the partitioned DCT.

Based on the idea of temporal data partitioning, we define N subsets s_i (with $i = 0, \dots, N-1$) of coefficients such that

$$\bigcup_{i=0}^{N-1} s_i = S, \quad (6)$$

where the set S contains all the 64 DCT coefficients. The subsets s_i are used to build up functions f_i that compute a scaled DCT for the coefficients in s_i . The functions f_i are applied to blocks with static contents in cyclical sequence (one per intercoded frame). After N intercoded frames, each coefficient for these blocks is computed at least once.

We set up an experiment using the “table tennis” sequence as follows in order to measure the effect of dynamic interframe coding. The computation of the DCT (for in-

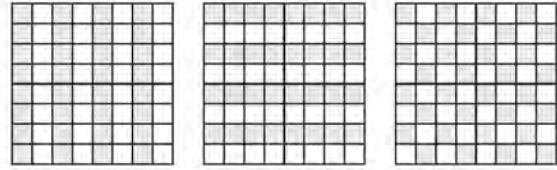


FIGURE 18: Example of coefficient subsets (marked gray) used for dynamic interframe DCT coding with a limitation to 32 coefficients per subset.

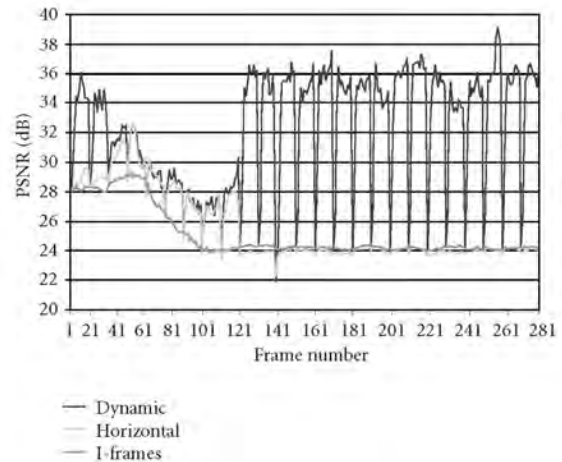


FIGURE 19: PSNR measures for the coded “table tennis” sequence, where the DCT computation was scaled to compute 32 coefficients. Compared to coding I-frames only (medium gray curve), inter DCT coding results in an improved output quality in case of motion (light gray curve) and even a higher output quality with dynamic interframe DCT computation.

traframe coding and interframe coding) was limited to 32 coefficients. The coefficient subsets we used are shown in Figure 18. Figure 19 shows the improvement in the PSNR that is achieved with this approach. Three curves are shown in this figure, plotting the achieved PSNR of the coded frames. The medium gray curve results from coding all the frames as I-frames, which we take as a reference in this experiment. The other two curves result from applying a GOP structure with $N = 16$ and $M = 4$. First, all blocks are processed with a fixed DCT (light gray curve) computing only the coefficients as shown in the left subset of Figure 18. It can be seen that when the content of the sequence changes due to movement, the PSNR increases. Second, the dynamic inter-DCT coding technique is applied to the coding process, which results in the dark gray curve. The dark gray curve shows an improvement to the light gray curve in case of no motion. The comb-like structure of the curve results from the periodic I-frame occurrence that restarts the quality buildup. The low periodicity of the quality drop gives a visually annoying effect that can be solved by computing more

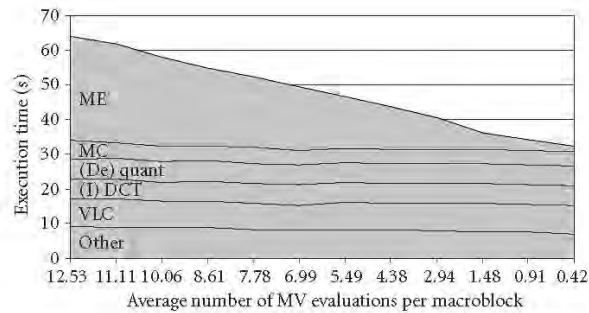


FIGURE 20: Example of ME scalability for the complete encoder when using a (12,4)-GOP ("IBBBP" structure) for coding.

coefficients for the I-frames. Although this seems interesting, this was not further pursued because of limited time.

5.5. Effect of scalable ME

The execution time of the MPEG modules when coding the "Stefan" sequence and scaling the ME is visualized in Figure 20. It can be seen that the curve for the ME block scales linearly with the number of MV evaluations, whereas the other processing blocks remain constant. The average number of vector candidates that are evaluated per macroblock by the scalable ME in this experiment is between 0.42 and 12.53. This number is clearly below the achieved average number of candidates (21.77) when using the diamond search [15]. At the same time, we found that our scalable codec results in a higher quality of the MC frame (up to 25.22 dB PSNR in average) than the diamond search (22.53 dB PSNR in average), which enables higher compression ratios (see the next section).

5.6. Combined effect of scalable DCT and scalable ME

In this section, we combine the scalable ME and DCT in the MPEG encoder and apply the scalability rules for (de)quantization, IDCT, and VLC, as we have described them in Section 2. Since the DCT and ME are the main sources for scalability, we will focus on the tradeoff between MVs and the number of computed coefficients.

Figure 21 portrays the obtained average PSNR of the coded "Stefan" sequence (CIF resolution) and Figure 22 shows the achieved bit rate corresponding to Figure 21. The experiments are performed with a (12,4)-GOP and $qscale = 1$. Both figures indicate the large design space that is available with the scalable encoder without quantization and open-loop control. The horizontally oriented curves refer to a fixed number of DCT coefficients (e.g., 8, 16, 24, 32, ..., 64), whereas vertically oriented curves refer to a fixed number of MV candidates. A normal codec would compute all the 64 coefficients and would therefore operate on the top horizontal curve of the graph. The figures should be jointly evaluated. Under the above-mentioned measurement conditions, the potential benefit of the scalable ME is only visible in the

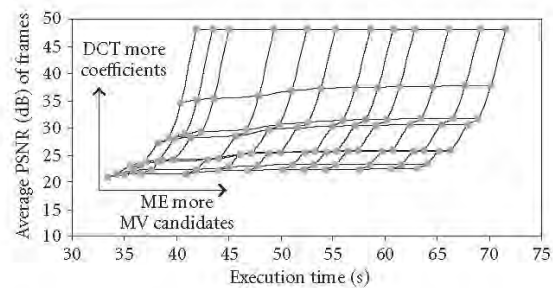


FIGURE 21: PSNR results of different configurations for the scalable MPEG modules.

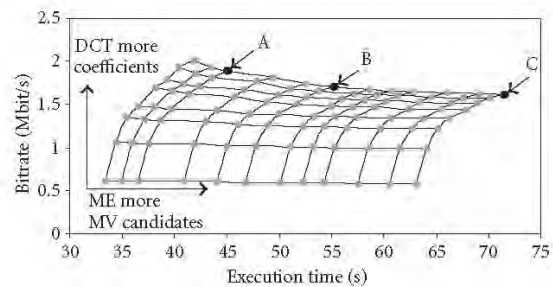


FIGURE 22: Obtained bit rates of different configurations for the scalable modules. The markers refer to points in the design space, where the same bit rate and quality (not computational complexity) is obtained as resulting from using diamond search (A) or full search with a 32×32 (B) or 64×64 (C) search area for ME.

reduction of the bit rate (see Figure 22) since an improved ME leads to less DCT coefficients for coding the difference signal after the MC in the MPEG loop.

In Figure 22, it can be seen that the bit rate decreases when computing more MV candidates (going to the right). The reduction is only visible when the bit rate is high enough. For comparison, the markers "A," "B," and "C" refer to three points from the design space. With these markers, the obtained bit rate of the scalable encoder is compared to the encoder using another ME algorithm. Marker "A" refers to the configuration of the encoder using the scalable ME, where the same bit rate and video quality (not the computational complexity) are achieved compared to the diamond search. As mentioned earlier, the diamond search performs 21.77 MV candidates on the average per macroblock. Our scalable coder operating under the same quality and bit rate combination as the diamond search in marker "A" results in 10.06 average MV candidates, thus 53.8% less than the diamond search. Markers "B" and "C" result from using the full-search ME with a 32×32 and 64×64 search area, respectively, requiring substantially more vector candidates (1024 and 4096, respectively). Figure 21 shows a corresponding measurement with the average PSNR, as the outcome, instead of the bit rate.

Figures 21 and 22 both present a large design space, but in practice, this is limited due to the quantization and bit rate control. Further experiments using quantization and bit rate control at 1500 kbps for the "Stefan," "Foreman," and "table tennis" sequence resulted in a quality level range from roughly 22 dB to 38 dB. As could be expected from inserting the quantization, the curves moved to lower PSNR (the lower half of Figure 21) and less computation time is required since fewer coefficients are computed. It was found that the remaining design space is larger for sequences having less motion.

6. CONCLUSIONS

We have presented techniques for complexity scalable MPEG encoding that gradually reduce the quality as a function of limited resources. The techniques involve modifications to the encoder modules in order to pursue scalable complexity and/or quality. Special attention has been paid to exploiting a scalable DCT and ME because they represent two computational expensive corner stones of MPEG encoding. The introduced new techniques for the scalability of the two functions show considerable savings of computational complexity for video applications having low-quality requirements. In addition, a scalable block classification technique has been presented, which is designed to support the scalable processing of the DCT and ME. In the second step, performance evaluations have been carried out by constructing a complete MPEG encoding system in order to show the design space that is achieved with the scalability techniques. It has been shown that even a higher reduction in computational complexity of the system could be obtained if available data (e.g., which DCT coefficients are computed during a scalable DCT computation) is exploited to optimize other core functions.

The obtained execution times of the encoder when coding the "Stefan" sequence as an example for complexity has been measured. It was found that the overall execution time of the scalable encoder can be gradually reduced to roughly 50% of its original execution time. At the same time, the codec provides a wide range of video quality levels (roughly from 20 dB to 48 dB PSNR in average) and compression ratios (from 0.58 to 2.02 Mbps). Further experiments targeting a bit rate of 1500 kbps for the Stefan, Foreman, and table tennis sequence result in a quality level range from roughly 21.5 dB to 38.5 dB. Compared with the diamond search ME from literature which requires 21.77 MV candidates on the average per macroblock, our scalable coder operating under the same quality and bit rate combination uses 10.06 average MV candidates, thus 53.8% less than the diamond search.

Another result of our experiments is that the scalable DCT has an integrated coefficient selection function which may enable a quality increase during interframe coding. This phenomenon can lead to an MPEG encoder with a number of special DCTs with different selection functions, and this option should be considered for future work. This should

also include different scaling of the DCT for intra- and inter-frame coding. For scalable ME, future work should examine the scalability potentials of using various fixed and dynamic GOP structures, and of concentrating or limiting the ME to frame parts, whose content (could) have the current viewer focus.

REFERENCES

- [1] C. Hentschel, R. Braspenning, and M. Gabrani, "Scalable algorithms for media processing," in *IEEE International Conference on Image Processing (ICIP '01)*, vol. 3, pp. 342–345, Thessaloniki, Greece, October 2001.
- [2] R. Prasad and K. Ramkishor, "Efficient implementation of MPEG-4 video encoder on RISC core," in *IEEE International Conference on Consumer Electronics, Digest of Technical papers (ICCE '02)*, pp. 278–279, Los Angeles, Calif, USA, June 2002.
- [3] K. Lengwehasatit and A. Ortega, "DCT computation based on variable complexity fast approximations," in *Proc. IEEE International Conference of Image Processing (ICIP '98)*, vol. 3, pp. 95–99, Chicago, Ill, USA, October 1998.
- [4] S. Peng, "Complexity scalable video decoding via IDCT data pruning," in *International Conference on Consumer Electronics (ICCE '01)*, pp. 74–75, Los Angeles, Calif, USA, June 2001.
- [5] Y. Chen, Z. Zhong, T. H. Lan, S. Peng, and K. van Zon, "Regulated complexity scalable MPEG-2 video decoding for media processors," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 8, pp. 678–687, 2002.
- [6] R. Braspenning, G. de Haan, and C. Hentschel, "Complexity scalable motion estimation," in *Proc. of SPIE: Visual Communications and Image Processing 2002*, vol. 4671, pp. 442–453, San Jose, Calif, USA, 2002.
- [7] S. Mientens, P. H. N. de With, and C. Hentschel, "New DCT computation technique based on scalable resources," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 34, no. 3, pp. 189–201, 2003.
- [8] S. Mientens, P. H. N. de With, and C. Hentschel, "Frame reordered multi-temporal motion estimation for scalable MPEG," in *Proc. 23rd International Symposium on Information Theory in the Benelux*, Louvain-la-Neuve, Belgium, May 2002.
- [9] N. Cho and S. Lee, "Fast algorithm and implementation of 2-D discrete cosine transform," *IEEE Trans. Circuits and Systems*, vol. 38, no. 3, pp. 297–305, 1991.
- [10] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Transactions of the Institute of Electronics, Information and Communication Engineers*, vol. 71, no. 11, pp. 1095–1097, 1988.
- [11] D. Farin, N. Mache, and P. H. N. de With, "A software-based high-quality MPEG-2 encoder employing scene change detection and adaptive quantization," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 4, pp. 887–897, 2002.
- [12] T. Kummerow and P. Mohr, *Method of determining motion vectors for the transmission of digital picture information*, EPO 496 051, European Patent Application, November 1991.
- [13] M. Chen, L. Chen, and T. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 5, pp. 504–509, 1994.
- [14] R. Li, B. Zeng, and M. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, 1994.
- [15] J. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 369–377, 1998.

- [16] P. N. H. de With, "A simple recursive motion estimation technique for compression of HDTV signals," in *IEEE 4th International Conference on Image Processing and Its Applications (IPA '92)*, pp. 417–420, Maastricht, The Netherlands, April 1992.
- [17] F. Rovati, D. Pau, E. Piccinelli, L. Pezzoni, and J. M. Bard, "An innovative, high quality and search window independent motion estimation algorithm and architecture for MPEG-2 encoding," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 697–705, 2000.
- [18] S. Mietens, P. H. N. de With, and C. Hentschel, "Computational complexity scalable motion estimation for mobile MPEG encoding," *IEEE Transactions on Consumer Electronics*, 2002/2003.

Stephan Mietens was born in Frankfurt (Main), Germany in 1972. He graduated in Computer Science from the Technical University of Darmstadt, Germany, in 1998 on the topic of "asynchronous VLSI design." Subsequently, he joined the University of Mannheim, where he started his research on "flexible video coding and architectures" in cooperation with Philips Research Laboratories in Eindhoven, The Netherlands. He joined the Eindhoven University of Technology in Eindhoven, The Netherlands, in 2000, where he is working towards a Ph.D. degree on "scalable video systems." Since 2003, he became a Scientific Researcher at Philips Research Labs. in the Storage and System Applications group, where he is involved in projects to develop new coding techniques.



Peter H. N. de With obtained his M.S. engineering degree from the University of Technology in Eindhoven in 1984 and his Ph.D. degree from the University of Technology Delft, The Netherlands in 1992. From 1984 to 1993, he joined the Magnetic Recording Systems Department, Philips Research Labs. in Eindhoven, and was involved in several European projects on SDTV and HDTV recording. He also contributed as a principal coding expert to the DV digital camcording standard. In 1994, he joined the TV Systems group, where he was leading advanced programmable architectures design as Senior TV Systems Architect. In 1997, he became a Full Professor at the University of Mannheim, Germany, in the Faculty of Computer Engineering. In 2000, he joined CMG Eindhoven as a principal consultant and he became a Professor in Electrical Engineering Faculty, University of Technology Eindhoven (EE Faculty). He has written numerous papers on video coding, architectures, and their realization. He is a Regular Teacher of postacademic courses at external locations. In 1995 and 2000, he coauthored papers that received the IEEE CES Transactions Paper Award. In 1996, he obtained a company Invention Award. Mr. de With is an IEEE Senior Member, Program Member of the IEEE CES (Tutorial Chair, Program Chair) and Chairman of the Benelux Information Theory Community.



Christian Hentschel received his Dr.-Ing. (Ph.D.) in 1989 and Dr.-Ing. habil. in 1996 from Braunschweig University of Technology, Germany. He worked on digital video signal processing with focus on quality improvement. In 1995, he joined Philips Research Labs. in Briarcliff Manor, USA, where he headed a research project on moiré analysis and suppression for CRT-based displays. In 1997, he moved to Philips Research Labs. in Eindhoven, The Netherlands, leading a cluster for programmable video architectures. He got the position of a Principal Scientist and coordinated a project on scalable media processing with dynamic resource control between different research laboratories. Since August 2003, he is a Full Professor at the University of Technology in Cottbus, Germany, where he heads the Department of Media Technology. He is a member of the Technical Committee of the International Conference on Consumer Electronics (IEEE) and a member of the FKTG in Germany.



Ex. 1018A



HOME

DATABASES

FIND JOURNALS

AI SEARCH
ASSISTANT

Sign in

Menu



JOURNAL

EURASIP journal on advances in signal processing.

European Association for Signal Processing, issuing body.; Hindawi Publishing Corporation, publisher.

2007



PEER REVIEWED



OPEN ACCESS

[Available Online \(Coverage: From 2001\) >](#)

TOP

VIEW ONLINE

SEND TO

SEARCH INSIDE

DOAJ Directory of Open Access Journals (WRLC)



DETAILED RECORD

Available from 2001.

VIRTUAL BROWSE

ProQuest Central

SHOW LICENSE



RELATED LINKS

Available from 12/01/2001 through 12/31/2004.
Available from 01/01/2007.

Springer Nature OA Free Journals (WRLC)



Available from 12/01/2001 volume: 2001 issue: 1.

Send to



EMAIL



PRINT



PERMALINK



CITATION



ENDNOTE



EASYBIB



MENDELEY



EXPORT TO
EXCEL



EXPORT RIS



EXPORT BIBTEX

Having trouble with this resource?

[Report a Problem](#)

Search inside

Search for articles within this journal

Article title or keyword



Detailed Record

Contributor	European Association for Signal Processing, issuing body. > Hindawi Publishing Corporation, publisher. >
Title	EURASIP journal on advances in signal processing.
Publisher	Heidelberg : SpringerOpen Began with vol. 2007 (2007).
Publication Date	2007
Format	1 online resource
Subject	Signal processing -- Periodicals > Signal processing > Periodicals >
Other title	EURASIP JASP European Association for Signal Processing journal on advances in signal processing Advances in signal processing
Frequency	Continuously updated
Language	English
Note	Refereed/Peer-reviewed All issues originally published with the title: EURASIP journal on applied signal processing, have been reformatted and issued under the later title: EURASIP journal on advances in signal processing.
Restrictions on Access	Unrestricted online access
Identifier	LC : 2008249018 ISSN : 1687-6180 OCLC : (OCoLC)133425278 ISSN : 1687-6172
Source	Library Catalog

Virtual Browse



Related Links

- [Search in WorldCat](#)  
- [Display Source Record](#)  

Ex. 1018B



EURASIP Journal on Advances in Signal Processing

Publishing model: [Open access](#)

CATHOLIC UNIV OF AMERICA MULLEN LIBRARY

[Explore open access funding](#)

[Change institution](#)

Journal menu

Volumes and issues

[Search all EURASIP Journal on Advances in Signal Processing articles](#) →

Volume 2025

Dec 2025

[Issue 1](#) | December 2025

Volume 2024

Dec 2024

[Issue 1](#) | December 2024

Volume 2023

Dec 2023

[Issue 1](#) | December 2023

Volume 2022

Dec 2022

[Issue 1](#) | December 2022

Volume 2021

Dec 2021

[Issue 1](#) | December 2021

Volume 2020

Dec 2020

[Issue 1](#) | December 2020

Volume 2019

Dec 2019

[Issue 1](#) | December 2019

Volume 2018

Dec 2018

[Issue 1](#) | December 2018

Volume 2017

Dec 2017

[Issue 1](#) | December 2017

Volume 2016

Dec 2016

[Issue 1](#) | December 2016

Volume 2015

Dec 2015

[Issue 1](#) | December 2015

Volume 2014

Dec 2014

[Issue 1](#) | December 2014

Volume 2013

Dec 2013

[Issue 1](#) | December 2013

Volume 2012

Dec 2012

[Issue 1](#) | December 2012

Volume 2011

Dec 2011

[Issue 1](#) | December 2011

Volume 2010

Dec 2010

[Issue 1](#) | December 2010

Volume 2009

Dec 2009

[Issue 1](#) | December 2009

Volume 2008

Dec 2008

[Issue 1](#) | December 2008

Volume 2007

Dec 2007

[Issue 1](#) | December 2007

Volume 2006

Dec 2006

[Issue 1](#) | December 2006

Volume 2005

Dec 2005

[Issue 20](#) | December 2005

[Issue 19](#) | December 2005

[Issue 18](#) | December 2005

[Issue 17](#) | December 2005

[Issue 16](#) | December 2005

[Issue 15](#) | December 2005

[Issue 14](#) | December 2005

[Issue 13](#) | December 2005

[Issue 12](#) | December 2005

[Issue 11](#) | December 2005

[Issue 10](#) | December 2005

[Issue 9](#) | December 2005

[Issue 8](#) | December 2005

[Issue 7](#) | December 2005

[Issue 6](#) | December 2005

[Issue 5](#) | December 2005

[Issue 4](#) | December 2005

[Issue 3](#) | December 2005

[Issue 2](#) | December 2005

[Issue 1](#) | December 2005

Volume 2004

Dec 2004

[Issue 17](#) | December 2004

[Issue 16](#) | December 2004

[Issue 15](#) | December 2004

[Issue 14](#) | December 2004

[Issue 13](#) | December 2004

[Issue 12](#) | December 2004

[Issue 11](#) | December 2004

[Issue 10](#) | December 2004

[Issue 9](#) | December 2004

[Issue 8](#) | December 2004

[Issue 7](#) | December 2004

[Issue 6](#) | December 2004

[Issue 5](#) | December 2004

[Issue 4](#) | December 2004

[Issue 3](#) | December 2004

[Issue 2](#) | December 2004

[Issue 1](#) | December 2004

Volume 2003

Dec 2003

[Issue 13](#) | December 2003

[Issue 12](#) | December 2003

[Issue 11](#) | December 2003

[Issue 10](#) | December 2003

[Issue 9](#) | December 2003

[Issue 8](#) | December 2003

[Issue 7](#) | December 2003

[Issue 6](#) | December 2003

[Issue 5](#) | December 2003

[Issue 4](#) | December 2003

[Issue 3](#) | December 2003

[Issue 2](#) | December 2003

[Issue 1](#) | December 2003

Volume 2002

Dec 2002 – Dec 2003

[Issue 12](#) | December 2003

[Issue 11](#) | December 2002

[Issue 10](#) | December 2002

[Issue 9](#) | December 2002

[Issue 8](#) | December 2002

[Issue 7](#) | December 2002

[Issue 6](#) | December 2002

[Issue 5](#) | December 2002

[Issue 4](#) | December 2002

[Issue 3](#) | December 2002

[Issue 2](#) | December 2002

[Issue 1](#) | December 2002

Volume 2001

Dec 2001

[Issue 4](#) | December 2001

[Issue 3](#) | December 2001

[Issue 2](#) | December 2001

[Issue 1](#) | December 2001

[Home](#) [EURASIP Journal on Advances in Signal Processing](#) [Volumes and issues](#) [Volume 2004, Issue 2](#)**EURASIP Journal on Advances in Signal Processing**Publishing model: [Open access](#)

CATHOLIC UNIV OF AMERICA MULLEN LIBRARY

[Explore open access funding](#)[Change institution](#)

Journal menu

[Search all EURASIP Journal on Advances in Signal Processing articles](#) →**Volume 2004, Issue 2**

December 2004

13 articles in this issue

Medusa: A Novel Stream-Scheduling Scheme for Parallel Video Servers

Research Article | 04 March 2004 | Article: 967232



Part of 1 collection:

[Multimedia over IP and Wireless Networks](#)**New Complexity Scalable MPEG Encoding Techniques for Mobile Applications**

Research Article | 04 March 2004 | Article: 785130



Part of 1 collection:

Multimedia over IP and Wireless Networks

Fine-Grained Rate Shaping for Video Streaming over Wireless Networks

Research Article 04 March 2004 Article: 637214



Part of 1 collection:

Multimedia over IP and Wireless Networks

Scalable Video Transcoding for the Wireless Internet

Research Article 04 March 2004 Article: 561576



Part of 1 collection:

Multimedia over IP and Wireless Networks

SMART: An Efficient, Scalable, and Robust Streaming Video System

Research Article 04 March 2004 Article: 421079



Part of 1 collection:

Multimedia over IP and Wireless Networks

Error Resilient Video Compression Using Behavior Models

Research Article 04 March 2004 Article: 412926



Part of 1 collection:

Multimedia over IP and Wireless Networks

Source and Channel Adaptive Rate Control for Multicast Layered Video Transmission Based on a Clustering Algorithm

Research Article 04 March 2004 Article: 396234



Part of 1 collection:

Multimedia over IP and Wireless Networks

An Integrated Source and Channel Rate Allocation Scheme for Robust Video Coding and Transmission over Wireless Channels

Research Article 04 March 2004 Article: 342539



Part of 1 collection:

Multimedia over IP and Wireless Networks

Interactive Video Coding and Transmission over Heterogeneous Wired-to-Wireless IP Networks Using an Edge Proxy

Research Article 04 March 2004 Article: 230385



Part of 1 collection:

Multimedia over IP and Wireless Networks

Performance and Complexity Co-evaluation of the Advanced Video Coding Standard for Cost-Effective Multimedia Communications

Research Article 04 March 2004 Article: 214371



Part of 1 collection:

Multimedia over IP and Wireless Networks

Editorial

Research Article 04 March 2004 Article: 107917



Part of 1 collection:

Multimedia over IP and Wireless Networks

Effective Quality-of-Service Renegotiating Schemes for Streaming Video

Research Article 04 March 2004 Article: 102607



Part of 1 collection:

Multimedia over IP and Wireless Networks

Optimal Erasure Protection Assignment for Scalable Compressed Data with Small Channel Packets and Short Channel Codewords

Research Article 04 March 2004 Article: 101469



Part of 1 collection:

Multimedia over IP and Wireless Networks

New Complexity Scalable MPEG Encoding Techniques for Mobile Applications

Research Article Published: 04 March 2004

Volume 2004, article number 785130, (2004) [Cite this article](#)**EURASIP Journal on Advances in Signal
Processing**[Submit manuscript](#)[Stephan Mietens](#) , [Peter HN de With](#) & [Christian Hentschel](#)723 Accesses [Explore all metrics](#) →

Abstract

Complexity scalability offers the advantage of one-time design of video applications for a large product family, including mobile devices, without the need of redesigning the applications on the algorithmic level to meet the requirements of the different products. In this paper, we present complexity scalable MPEG encoding having core modules with modifications for scalability. The interdependencies of the scalable modules and the system performance are evaluated. Experimental results show scalability giving a smooth change in complexity and corresponding video quality. Scalability is basically achieved by varying the number of computed DCT coefficients and the number of evaluated motion vectors but other modules are designed such they scale with the previous parameters. In the experiments using the "Stefan" sequence, the elapsed execution time of the scalable encoder, reflecting the computational complexity, can be gradually reduced to roughly 50% of its original execution time. The video quality scales between 20 dB and 48 dB PSNR with unity quantizer setting, and between 21.5 dB

and 38.5 dB PSNR for different sequences targeting 1500 kbps. The implemented encoder and the scalability techniques can be successfully applied in mobile systems based on MPEG video compression.

 This is a preview of subscription content, [log in via an institution](#)  to check access.

Access this article

[Log in via an institution](#)

Subscribe and save

 Springer+ Basic

\$34.99 /Month

Get 10 units per month

Download Article/Chapter or eBook

1 Unit = 1 Article or 1 Chapter

Cancel anytime

[Subscribe now →](#)

Buy Now

[Buy article PDF USD 39.95](#)

Price excludes VAT (USA)

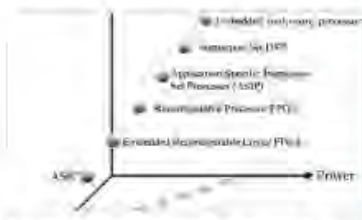
Tax calculation will be finalised during checkout.

Instant access to the full article PDF.

Rent this article via [DeepDyve](#) 

[Institutional subscriptions →](#)

Similar content being viewed by others



Survey on Algorithm and VLSI Architecture for MPEG-Like Video Coder

Article 05 September 2016



Parallel Integer Motion Estimation for High Efficiency Video Coding (HEVC) Using OpenCL

Chapter © 2017



A hardware-friendly algorithm for LCU-level pipe-lined integer motion estimation

Article 14 June 2023

Author information

Authors and Affiliations

Philips Research Laboratories, Prof. Holstlaan 4, Eindhoven, NL-5656 AA, The Netherlands
Stephan Mietens

LogicaCMG Eindhoven, Eindhoven University of Technology, P.O. Box 7089, Luchthavenweg 57,
Eindhoven, NL-5600 MB, The Netherlands
Peter HN de With

Cottbus University of Technology, Universitätsplatz 3-4, Cottbus, D-03044, Germany
Christian Hentschel

Corresponding author

Correspondence to [Stephan Mietens](#).

Rights and permissions

[Reprints and permissions](#)

About this article

Cite this article

Mietens, S., de With, P.H. & Hentschel, C. New Complexity Scalable MPEG Encoding Techniques for Mobile Applications. *EURASIP J. Adv. Signal Process.* **2004**, 785130 (2004). <https://doi.org/10.1155/S1110865704309091>

Received
10 December 2002

Revised
07 July 2003

Published
04 March 2004

DOI
<https://doi.org/10.1155/S1110865704309091>

Keywords and phrases

[MPEG encoding](#)

[scalable algorithms](#)

[resource scalability](#)

Client-Server Synchronization and Buffering for Variable Rate Multimedia Retrievals

Joseph Y. Hui, *Senior Member, IEEE*, Ezhan Karasan, *Student Member, IEEE*,
Jun Li, *Student Member, IEEE*, and Junbiao Zhang

Abstract—We consider the use of large buffers and feedback as a mechanism to maintain loosely coupled synchronization between a multimedia server and a client. The multimedia stream is modeled as a fluid flow through rate controlled valves and buffers with multiple thresholds. These thresholds are used to control the rates upstream. The quality of service for the multimedia connection is characterized in terms of the jitter in the received media stream due to buffer underflow and overflow. This quality of service is used to exercise rate and admission control in the presence of congestion. The feedback mechanism is implemented in GRAMS, an adaptive multimedia client-server system. Experimental statistics are gathered for the purpose of traffic engineering. We employ a fluid flow and first passage time analysis to understand the traffic process through the pipelines and the buffers and to estimate the amount of signaling required by the feedback mechanism.

I. INTRODUCTION

ISSUES OF SYNCHRONIZATION AND BUFFERING FOR MULTIMEDIA

MULTIMEDIA communications, as the name implies, involve media of different types such as video, audio, image, text, and data. These communication sessions could be further classified according to their requirements for synchronization and delay. First, real time communications, such as teleconferencing, require the strictest synchronization and the least time differences among the time instants of signal capturing, signal coding, signal buffering, signal transmission, signal reception, and signal display. The end-to-end delay should typically be less than tens of milliseconds. Second, retrieval of multimedia information requires only the display of the audio-video at the same rate as its capturing. The delay between signal retrieval and signal display could be as large as several seconds. However, larger delay may result in unsatisfactory response time for the client. Also, a large delay may require the use of a large buffer somewhere between the server and the client display. Third, image transport imposes almost no synchronization requirement among the components of the signal, since the image typically is displayed upon

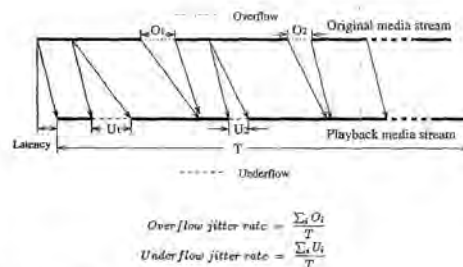


Fig. 1. Jitter rate related to underflow and overflow.

the completion of the entire image transmission. The delay of concern here is the delay between the initiation of an image request, until the complete image is received. This delay should ideally be less than a second, but delay less than 10 ms may be of no subjective advantage to the client.

We believe that delay is a significant measure of the quality of service (QoS) for multimedia. Here, we distinguish between two kinds of delay, latency versus jitter. Latency, defined for video, audio, and image transport, is the delay between signal generation (or retrieval) and signal display. Jitter, defined for continuous media streams such as audio and video, results from the disruption of the media as a consequence of underflow or overflow conditions (Fig. 1). The underflow jitter rate is defined as the fraction of time the received stream is paused due to underflowing. The overflow jitter rate is defined as the fraction of time the received stream is lost due to overflowing. The sum of these two rates is termed as the jitter rate, which measures how much jitter there is in the media stream. Jitter is due largely to the failure to synchronize the server and the client in order to maintain continuous flow. Such synchronization is achieved, in this paper, through buffer occupancy feedback.

As the current network technology supports the means to handle multimedia traffic and the user demand for multimedia services increases, protocols are required to provide satisfactory quality of service for multimedia communications. Synchronization, rate control, and buffer management between the source and the destination are three closely related problems to be solved in order to provide satisfactory quality of service for multimedia communication [1]–[11]. Different approaches have been proposed to address the problem of synchronization: models based on Petri nets [2], [4], [8],

Manuscript received August 8, 1994; revised March 30, 1995. This work was supported by the National Science Foundation under contract NSF NCR 90-58079-02 and with matching funds from Bellcore and NEC.

J. Y. Hui, E. Karasan, and J. Li are with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08855 USA.

J. Zhang is with the Department of Computer Science, Rutgers University, Piscataway, NJ 08855 USA.

Publisher Item Identifier S 0733-8716(96)00243-0.

0733-8716/96\$05.00 © 1996 IEEE

feedback techniques for synchronizing distributed media [6], and schemes providing synchronization at the buffer level [11].

In this paper, we shall devise feedback mechanisms for the flow control and synchronization problems, analyze the stochastic behavior of the resulting problem of media data flow with feedback, and give empirical traffic data from the resulting implementation.

In particular, we use large buffers with three threshold levels to maintain loosely coupled synchronization between a multimedia server and a client. These thresholds are used to produce feedback information to exhibit rate control for preventing buffer underflow and overflow. The quality of service for the multimedia connection is expressed in terms of the jitter in the received media stream due to underflow and overflow, given in Fig. 1. This synchronization protocol is implemented in gopher-style real time ATM multimedia system (GRAMS) [12], a client-server system for multimedia communication. We characterize the resulting traffic process by using a first passage time analysis, which is used to predict the frequency of signaling required to implement the feedback mechanism.

In the multimedia server GRAMS [1], we implemented a client interface that provides client-based synchronization and QoS control via feedback. GRAMS is a distributed multimedia system that can be deployed on an ATM LAN, a wide area ATM network, or the Internet environment. The most noticeable features of this system are:

- GRAMS is fully compatible with several popular Internet document retrieval protocols including Gopher and http [17]. The client interface of GRAMS employs the popular media viewer Mosaic [18], which provides integrated multimedia display as well as the World-Wide Web (WWW) navigation help [17].
- The GRAMS server can provide heterogeneous media services such as video, audio, image, and text with QoS control.
- Several video formats (including MPEG, motion JPEG [13] and CELLB [19]) are supported in addition to image formats such as JPEG and GIF.
- GRAMS supports multiple transport protocols including TCP, UDP and ATM with AAL4 [20].

The implementation of the multimedia server in GRAMS is explained in Appendix A. Interested readers may refer to [12] for further details.

The rest of the paper is organized as follows. In Section II, we explain the feedback flow control and synchronization paradigm. The details of the flow control scheme based on three buffer thresholds are presented. In Section III, media scaling and synchronization in GRAMS are described. In Section IV, experimental results are given for the buffer occupancy and signaling frequency, obtained by using a video sequence encoded by CELLB [19]. In Section V, we analyze the traffic rate process resulting from the feedback rate control scheme. In particular, we obtain the mean time spent using a particular rate until rate changes due to feedback from the receiver end. We calculate the proportion of time each rate is used, that is used to obtain the frequency of signaling

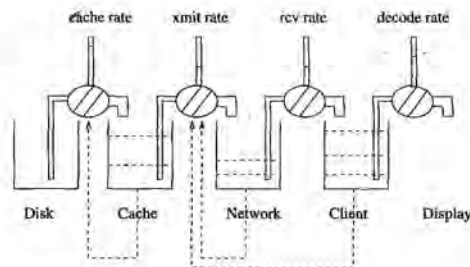


Fig. 2. Client-based rate and QoS control.

information from the client to the server. Section VI concludes the paper.

II. FEEDBACK FLOW CONTROL AND SYNCHRONIZATION PARADIGM

In the GRAMS implementation, the media stream from the server to the client may be modeled as fluid flow through a number of valves and buffers as shown in Fig. 2. There are essentially four buffers where the media stream may reside temporarily. First, the media stream may be stored in a hard disk. For teleconferencing, the media stream may be generated by the capture board, and subsequently compressed by media coding algorithms. Second, these streams are then cached at the transmit buffer. Third, the network, due to its transit latency and buffers within the network, may be viewed as another buffer. Fourth, the receive end may buffer a sufficient amount of the media stream for the purposes of maintaining a continuous stream for display and suitable synchronization with the transmit end.

Connecting these buffers are variable rate valves, some of which could be controlled using the state information of one or more of the previously described buffers. For stored video retrieval, the transfer rate from the hard disk to the system cache is dictated by the constant retrieval speed of the bus of the hard drive, even though the media stream may be of variable rate. For teleconferencing, the transfer rate could be highly variable as a function of the type of video frame (I, P and B frames for MPEG [13]) as well as the amount of motion and the granularity of the video. Some rate smoothing is necessary via the use of the transmit buffer. The transmit valve is probably the most critical control in the pipeline, performing the vital traffic shaping function. The receive valve on the client end ideally should receive information at the same rate as the transmit valve. A small transit latency may be experienced. The receive rate may match the transmit rate closely by adequate allocation of buffer and bandwidth resources [14] within the network. The video display valve empties data from the client buffer at a rate that maintains a smooth playback.

Transmitter-receiver synchronization is achieved through flow rate modulation. The control of rates as a function of buffer occupancy can be of two kinds. The first kind, that receives more attention in the literature [15], [16], consists of feedforward control, namely that the upstream buffer oc-

cupancy determines the rates of the valves downstream. For example, a high occupancy of the transmit buffer may result in a feedforward increase in the transmit rate and the display rate. A high transmit buffer occupancy could result from a surge in the media coding rate, such as, that resulting from motion in the video.

Feedforward control is often used for teleconferencing for which delay should be less than tens of milliseconds. Subsequently, the buffers in the end-to-end connection should be small. As a result, control actions may be frequent. Such frequent adjustments within the network could be difficult to achieve and often bandwidth overengineering is used to minimize the effect of rate surges. Such engineering requires certain knowledge of traffic statistics.

The focus of this paper is on the second kind of control, that consists of feedback control [5], namely the downstream buffer occupancy determines the rates of the valves. For example, a high occupancy of the receive buffer indicates that the transmit rate is too high compared to the video display bit rate. This low display rate could result from an inherently low video coding rate or the failure of the client to decode at the regular frame rate. The client then sends a feedback request to the server for a reduction in the transmit rate. Similarly, a low occupancy of the receive buffer may result in a feedback request from the client to the server for an increase in the transmit rate. A low occupancy of the receive buffer, however, can also result from a congested network, with data accumulating in the network buffer. In this circumstance, increasing the transmit rate would be detrimental. This condition should be detected by a mismatch between the requested transmit rate and the receive rate. Congestion is then alleviated by measures such as lowering the frame rate for the video and subsequently, the transmit rate.

The advantages of feedback control are many. First, this protocol emphasizes client QoS control, taking into account that the receiver has better and immediate knowledge of the condition of the end-to-end connection, including the network condition which may not be readily accessible to the transmitter. In certain sense, the effect of all upstream congestion due to the server or the network is detected, differentiated, and alleviated by the action of the client. Second, the transmitter need not examine the timing relationship and data format of the media, thus reducing the computational load of the server that often serves tens and possibly hundreds of connections at the same time. Third, the feedback automatically takes care of client-server synchronization. Fourth, this feedback mechanism does not require a detailed description of the video coding rate process other than the mean rate and hence it is robust to errors in traffic models.

If a large buffer is used, the frequency of feedback actions is low, albeit at the expense of a large delay. This infrequent change of a constant requested rate allows careful and infrequent bandwidth reservation [14]. Substantial traffic averaging is achieved as a result. Hence feedback control is more suitable for retrieval of stored media, for which a larger delay between the transmitter and the receiver is allowed.

We describe our feedback flow control and synchronization paradigm for the GRAMS system as follows. Thresholds are

set for the server cache, the network buffer, and the client buffer. For the server cache, a cache request to the hard disk is generated when half of the information in the cache has been transmitted. The cache manager then schedules a disk read to fill up the emptied half of the cache. To detect the occupancy level of the network buffer, the client keeps track of the receiver deficit as an accumulation of the difference between the requested transmit rate and the received rate. If this deficit exceeds a certain threshold, we declare the network throughput a bottleneck and actions will be taken to reduce the frame rate for the session. Thus the client interface adjusts the media quality adaptively with respect to the transport capability of the underlying network.

The receive buffer occupancy is used as control information for maintaining client-server synchronization. There are three buffer thresholds T_h , T_m , T_l , namely, the high, medium, and low thresholds. When the buffer occupancy crosses any one of these thresholds, a new transmit rate λ_h , λ_m , λ_l is requested if that rate is not the current transmission rate. The medium rate λ_m is equal to the average rate of the media, which is easily found by dividing the size of the media file by the duration of the media display. The high rate should normally be higher than the peak rate, and the low rate should normally be lower than the lowest rate. These extremal rates are more difficult to obtain than the average rate, and very often they are assigned to be a certain factor of the average rate. To safeguard against errors in estimating the peak and lowest rate, overflow or underflow conditions may be used to adjust these rates. For example, an overflow may temporarily result in a requested reduction of transmit rate to zero. Repeated overflow may necessitate a decrease of λ_l . Repeated underflow may necessitate an increase of λ_h .

In the following sections, experimental and theoretical statistics are obtained for the resulting traffic processes and buffer occupancy.

III. MEDIA SCALING AND SYNCHRONIZATION IN GRAMS

As discussed earlier, consistent underflow of the client buffer could indicate a network bottleneck or temporary congestion. In such a situation, the client requests a scale down of the media, thus reducing the bit rate requirement. In this section, we focus on media scaling and inter-media synchronization in GRAMS.

In GRAMS, a typical application is transmission of stored lectures often involving the synchronization of the video (e.g., the lecture scene), audio (e.g., the lecturer's speech that often carries most of the cognitive information of the lecture), and image streams (e.g., lecture transparencies). We call this an audio-video-visual session. The audio-video synchronization requirement is very stringent with a discrepancy of less than 100 ms, whereas, the video-image synchronization requirement could be quite loose with an allowable discrepancy in the order of seconds. Therefore, the media synchronization discussed here will primarily be between the video and audio streams.

There are two components of audio-video synchronization. First, we maintain the natural rate of playback of audio, that

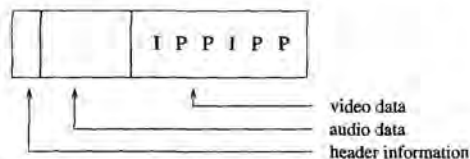


Fig. 3. Structure of the video-audio block.

from a subjective point of view is more important than the natural rate of playback of video [7]. Hence, we shall maintain the continuity of the audio, at the expense of possibly introducing jitter for the video. Second, we need to synchronize the audio stream with the video stream for qualities such as lip synchronization.

Many currently available technologies record and store the audio and video streams separately. This complicates the synchronization of the two media. In the GRAMS experiments, we group the streams into audio-video groups. Each group consists of six frames of video preceded by the audio data of the group. We assume the natural frame rate of 30 frames/s. For our current implementation, we use the following three video formats: motion JPEG that consists of I frames, MPEG that consists of periodic IPP frames, and CELLB. (The MPEG standard contains B frames besides I and P frames.) The number of frames in a group was chosen so that we may restart from an I frame, if packet loss or error occur during transmission of the group, or from an I frame of the next group. Also, the grouping allows scaling back of the video rate if the network is found to be a bandwidth bottleneck (Fig. 3).

In a separate pointer file, we store headers for each audio-video group. The header makes it easy for the server to find the I frames if for some reason it needs to decimate the video stream by dropping frames. The header contains the offset of the beginning and the end of the first and third frames, which are I frames for both motion JPEG and MPEG. The audio part precedes at the location indicated by the offset of the first frame. These offsets are transmitted in the header for the group and are used both by the server and the client during media transmission and playback.

Two features complicate the design of the audio-video synchronization scheme: media scalability and transport protocol support. These two features handle problems resulting from network bandwidth bottleneck, packet loss, and server-client overload.

As discussed earlier, the client detects network congestion and requests a downward scaling of the media to relieve the bandwidth bottleneck. Our grouping would allow frame rates of 30 frames/s, 10 frames/s (by transmitting the audio plus first and third I frames only), 5 frames/s (by transmitting the audio plus the first I frame), 2.5 frames/s (by transmitting the audio for every group and the video for every other group), as well as lower rates such as 1 frames/s, every two seconds, and every five seconds. The downward adjustment is continued until the requested transmission rate matches the receive rate. Upward frame rate adjustment is initiated by the end-user through a control on the client interface. We also note that caching is performed for the maximum frame rate even though only part

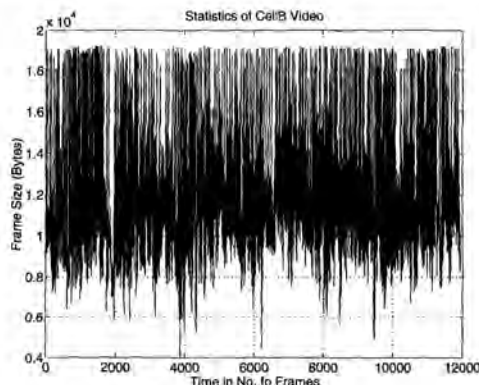


Fig. 4. A CELLB video sequence.

of the cached media may be transmitted. It is not difficult to cache only the transmitted information, but the increased disk seek time and address computations may outweigh the advantages of decreased disk reading time and reduced cache size.

This header structure devised for media scaling and synchronization is also used for accomplishing other functions of a transport protocol. The transmitter may employ a transport protocol such as TCP or UDP. Since image transport typically requires almost no packet loss and one or more seconds of image transport latency can be tolerated, using TCP as a transport mechanism could be appropriate. For audio-video applications, TCP may incur unnecessary re-transmission, delay, and processing. One option for audio-video transport is to add minimal transport layer functions within the application for packets piped via ATM and AAL 4/5 [20].

To handle out-of-order packets and packet loss, additional information is inserted into the packet header. For audio-video applications, packet loss may result in discarded or repeated audio-video segments. Currently, we do not attempt to re-transmit lost packets to avoid re-transmission delay and additional complexity in caching, re-transmission scheduling, and preventing packet duplication.

IV. EXPERIMENTAL RESULTS CONCERNING BUFFER SIZING AND RATE CONTROL

In this section, we present experimental traffic statistics as a result of client feedback based on receive buffer occupancy. The result is based on the video coding method CELLB, the currently available variable rate video encoding algorithm on our computing platform. The experiment is performed on a video stream for which a typical trajectory of the encoded size of frames is shown in Fig. 4. The video stream has the following statistics:

Number of frames $N = 12\,000$

Frame rate $R_f = 29.4$ frames/s

Bit rate $R_b = 334\,414$ bytes/s

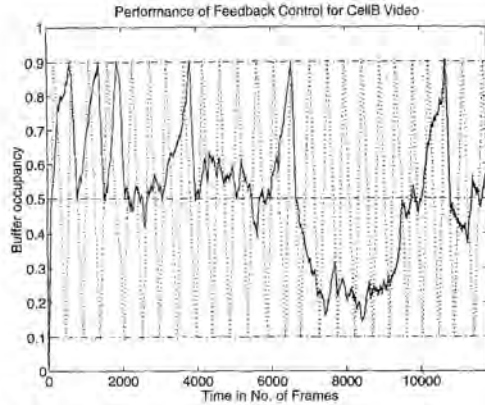


Fig. 5. Buffer occupancy variation for the CELLB video.

Statistics and modeling of variable rate MPEG can be found in [21]. Since we employ multiple audio-video standards as well as multiple sub-sampled frame rates, it seems from these rate trajectories that a proper modeling and adaptation of the model as a function of control could be quite complex. Our control approach is highly adaptive and does not require traffic statistics to be known *a-priori* other than the average rate (for the full rate and perhaps for the sub-sampled rate). Nevertheless, it is helpful to obtain traffic statistics of the requested rates by the client for the following purposes:

- Bandwidth engineering such as the sizing of bandwidth for a virtual path.
- Estimating the amount of signaling required due to these requests.
- Sizing the client buffer in order to maintain good QoS.

The choices of buffer size and thresholds affect the feedback frequency that indicates the overhead incurred by signaling and rate changes. Determining a proper combination of buffer size, thresholds and the corresponding transmission rates for a specific stream is very complicated due to many changing factors. In order to avoid overwhelming complexities, we set the three thresholds proportional to the buffer size and the three rates proportional to the average rate of the video stream. Tests are then performed to find out the buffer occupancy changes with respect to different buffer sizes. The actual values of the parameters are as follows

$$\begin{aligned}
 \text{Buffer size } S &= T * R_b \\
 \text{High Threshold } T_h &= 0.9 * S \\
 \text{Medium Threshold } T_m &= 0.5 * S \\
 \text{Low Threshold } T_l &= 0.1 * S \\
 \text{High rate } \lambda_h &= 1.5 * R_b \\
 \text{Medium rate } \lambda_m &= R_b \\
 \text{Low rate } \lambda_l &= 0.5 * R_b
 \end{aligned}$$

where T ranges from 1 s to 10 s in our tests.

Fig. 5 demonstrates a typical buffer occupancy as the client consumes the video stream obtained from the GRAMS server,

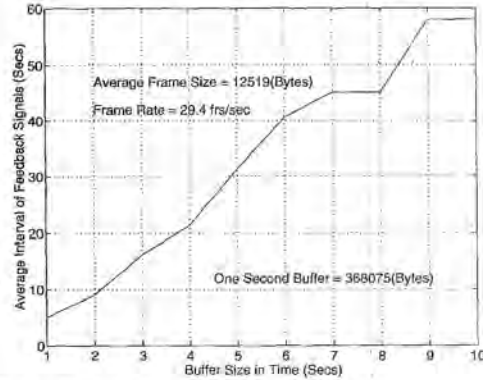


Fig. 6. Signaling frequencies relative to buffer size.

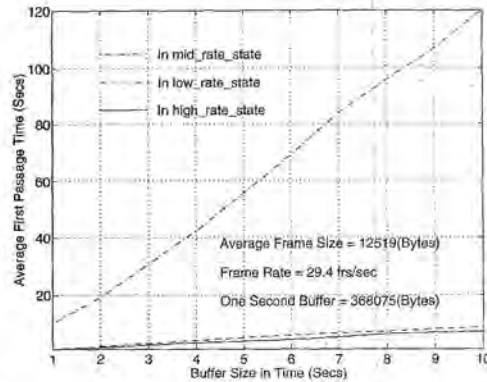


Fig. 7. First passage times relative to buffer size.

with $T = 5$ s. The curve in solid line shows the occupancy variations based on the feedback mechanism described earlier. The curve in dashed line gives the result when the medium threshold T_m is not used (i.e., the server only transmits at either the high rate λ_h or the low rate λ_l in response to client signals). In both cases, no buffer underflow or overflow occurs. We observe that the high threshold and low threshold are reached much less frequently when the medium threshold is set for the buffer. This will lead to a lower probability of buffer underflowing and overflowing. Rate changes are also reduced in this case.

Fig. 6 shows the change of average time interval between two consecutive feedback signals as the buffer size increases. The buffer size is given in the units of buffer space to store 1 s of video stream, called one second buffer (in this experiment one second buffer is approximately 368 kbytes). Even if we only use a buffer size corresponding to 1 s video stream, the average interval between rate changes is still larger than 5 s. Therefore, the overhead of feedback signaling is very small.

Fig. 7 shows the first passage times of the three transmit states. We distinguish three states according to the transmission

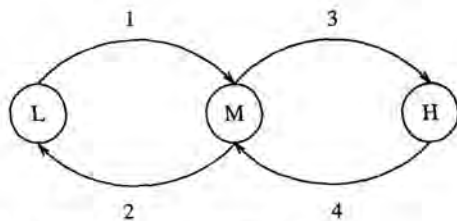


Fig. 8. FSM model for the transmission rate process.

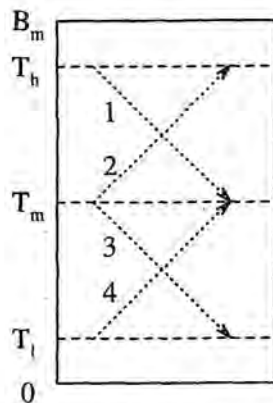


Fig. 9. Threshold crossings at the receive buffer.

rates the server chooses. The first passage time of each state denotes the average duration in which the server uses the corresponding rate. The curves show certain similarities to those in Fig. 10, that are obtained by using the analysis of the two-state Markov model, presented in the next section. However, the first passage time of medium rate state exhibits more linearity than exponentiality. Therefore, the ratio of the server using medium rate to low-high rate is almost unchanged with respect to the buffer size. The time the server spends in the medium rate state is about 4 times longer than the time spent in the low-high rate states.

These test results show the adaptiveness of the feedback mechanism in accommodating the fluctuations. The intuitive choices for the values of the thresholds and the rates are justified by the relatively smooth changes of the buffer occupancy and the transmit rates.

V. ANALYSIS OF THE RATE CONTROL PROCESS

In this section, we analyze the performance of the large receive buffer. In particular, we derive the mean duration between threshold crossings by using a first passage time analysis. The frequency of the rate change requests by the client to the server determines the amount of signaling required by the rate control protocol. The fraction of time the server is transmitting at each rate is also computed. Finally, a numerical example is presented for a two-state Markov chain model of the client decoding algorithm for VBR video service.

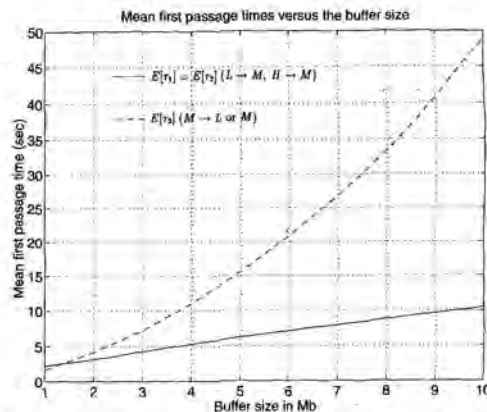


Fig. 10. Mean first passage times as a function of buffer size.

The choice of the three buffer thresholds as well as the transmission rates determine the frequency of rate change requests to maintain satisfactory performance for the multimedia service. For a fixed buffer size, if the thresholds are chosen at large distances from each other, the amount of signaling will be low at the expense of high overflow and underflow probability. The statistics of the rates used by the client decoding algorithm is another factor in selecting the thresholds. For example, for a VBR video application containing long periods of high and/or slow motion, the thresholds should be closer to prevent frequent buffer overflow and/or underflow. On the other hand, for a video sequence having bursts of high motion, closer thresholds will cause unnecessary signaling due to frequent threshold crossings.

In the following analysis, the multimedia arrival process is modeled as a fluid flow generated by a finite-state machine (FSM) having three states: Low(L), Medium(M) and High(H). When the FSM is in state L , M , or H , the arriving bit stream is a constant rate fluid flow with rate λ_l , λ_m , or λ_h , respectively, with $\lambda_l < \lambda_m < \lambda_h$. The state transition diagram of the FSM is given in Fig. 8 with each state transition labeled from one to four.

The client buffer has size B_M and three thresholds: T_l , T_m , and T_h with $T_l < T_m < T_h$. These thresholds are used by the client to produce feedback information to the server asking to change the rate of the multimedia flow to prevent buffer overflow and underflow. All possible threshold crossings are given in Fig. 9 with each transition labeled from one to four. Each threshold crossing in Fig. 9 has the same label with the corresponding state transition of the server transmission process given in Fig. 8.

The rate at which a user removes bits from the buffer is modeled by a fluid flow process modulated by a continuous time Markov chain with state transition rate matrix Q . The Markov modulated fluid flow model has been widely used to model the cell arrival process in ATM networks since its introduction in [22]. When the chain is in state s , information is removed at a rate R_s (in bits/s) from the buffer. The Markov

chain model captures the time correlation of rate processes generated by many multimedia sources such as VBR video encoders like MPEG. The Markov chain model of a rate process can be obtained by using the statistics of the process both in time and frequency domains. By looking the histogram of the rate process showing the frequency of each rate, one can identify clusters and assign a state to each cluster with the rate of the state equal to average rate of the cluster. The relative frequency of each cluster gives the steady state probability of the corresponding state. The time behavior of the rate process is used to obtain which transitions between clusters are possible. The average time rate stays within a cluster gives the sum of the transition rates out of the corresponding state.

In multimedia applications, a user may remove data from the buffer in batches at discrete times. As an example, an MPEG video decoder extracts each frame individually. If the thresholds are well separated, however, the time between threshold crossings will be much larger than the time between frame removals. For the computation of mean threshold crossing times, it is unnecessary to concentrate on the detailed behavior of frames. In the larger time scale, the bits are removed almost continuously. Thus, fluid flow model applies for the analysis of the events occurring in this time scale.

The following analysis can be incorporated into the client rate control protocol that adaptively changes the system parameters to achieve the desired performance during the course of a service. The client uses a Markov chain model to characterize the rate process. The parameters of the model are updated by monitoring the rate process through a time window. The buffer thresholds and transmission rates to attain a satisfactory performance are obtained by using the analysis. The transmission rates are periodically updated by sending the new values to the server.

A. First Passage Time Analysis

We analyze the expected time between threshold crossings. We assume that the feedback information is immediately available at the server. For local area networks, propagation delay is small compared to the mean period between rate changes and thus can be neglected.

We define the following three different first passage times:

- τ_1 is the first passage time from T_h to T_m (corresponding to state transition $L \rightarrow M$).
- τ_2 is the first passage time from T_l to T_m (corresponding to state transition $H \rightarrow M$).
- τ_3 is the first passage time from T_m to either T_h or T_l (corresponding to state transitions $M \rightarrow L$ or $M \rightarrow H$, respectively).

In order to find the mean values of τ_1 , τ_2 , and τ_3 , we define the following first passage time:

$$\tau = \min\{t : x(t) = B_1 \text{ or } x(t) = B_2 \text{ with } x(0) = x, S(t) = s, \text{ arrival rate } \lambda, B_1 < B_2\}$$

where $x(t)$ and $S(t)$ denote respectively the length of the buffer and the state of the Markov chain at time t . We define $E[\tau | s, x]$ as the expected value of τ starting at state (s, x) .

$E[\tau_i | s]$, $i = 1, 2, 3$ can be computed from $E[\tau | s, x]$ as follows:

- $E[\tau_1 | s] = E[\tau | s, T_h]$ with $B_1 = T_m$, $B_2 = \infty$ and $\lambda = \lambda_l$.
- $E[\tau_2 | s] = E[\tau | s, T_l]$ with $B_1 = -\infty$, $B_2 = T_m$ and $\lambda = \lambda_h$.
- $E[\tau_3 | s] = E[\tau | s, T_m]$ with $B_1 = T_l$, $B_2 = T_h$ and $\lambda = \lambda_m$.

$E[\tau | s, x] = 0$ for $x \geq B_2$ and $x \leq B_1$. For $B_1 < x < B_2$ and starting at a state s with $\lambda > R_s$, the queue length will hit the upper threshold B_2 if the Markov chain stays at s for longer than $(B_2 - x)/(\lambda - R_s)$. If the state changes earlier, $E[\tau | s, x]$ can be written recursively

$$E[\tau | s, x] = e^{-q_s \frac{B_2 - x}{\lambda - R_s}} \frac{B_2 - x}{\lambda - R_s} + \sum_{s' \neq s} q_{ss'} \int_0^{\frac{B_2 - x}{\lambda - R_s}} e^{-q_s t} \cdot [t + E[\tau | s', x + (\lambda - R_s)t]] dt \quad (1)$$

where $q_{ss'}$ is the transition rate from s to s' and $q_s = \sum_{s' \neq s} q_{ss'}$ is the rate of the exponential distribution governing the time spent at s .

When the chain starts at a state s with $\lambda < R_s$, the buffer length will hit the lower threshold B_1 if no transition occurs from s for a duration of at least $(x - B_1)/(R_s - \lambda)$ and $E[\tau | s, x]$ can be written recursively as

$$E[\tau | s, x] = e^{-q_s \frac{x - B_1}{R_s - \lambda}} \frac{x - B_1}{R_s - \lambda} + \sum_{s' \neq s} q_{ss'} \int_0^{\frac{x - B_1}{R_s - \lambda}} e^{-q_s t} \cdot [t + E[\tau | s', x + (\lambda - R_s)t]] dt. \quad (2)$$

We define the Laplace transform $F_s(\theta)$ of $E[\tau | s, x]$ with respect to the initial buffer length as

$$F_s(\theta) = \int_0^\infty E[\tau | s, x] e^{-\theta x} dx \quad (3)$$

and define the N -dimensional vector

$$\underline{F}(\theta) = [F_1(\theta), \dots, F_N(\theta)]^T \quad (4)$$

where N is the number of states in the Markov chain. We prove in Appendix B that $\underline{F}(\theta)$ can be written as

$$\underline{F}(\theta) = (Q + \theta D)^{-1} \underline{b}(\theta) \quad (5)$$

where $D = \text{diag}\{\lambda - R_s\}$ is the drift matrix and $\underline{b}(\theta)$ as defined in Appendix B is an N -dimensional vector. With the rate control process, the probability of buffer overflow and underflow will be very small and hence the boundary conditions at zero and B_M are neglected in the analysis.

The inverse Laplace transform can be performed by using partial fraction expansion. As shown in Appendix B, we can write

$$\underline{b}(\theta) = e^{-\theta B_1} \underline{b}_1 + e^{-\theta B_2} \underline{b}_2$$

where b_1 and b_2 are obtained by applying boundary conditions at B_1 and B_2 . Let $\{\theta_k\}_{k=1}^N$ be the roots of the equation

$$\det(Q + \theta D) = 0. \quad (6)$$

Since all row sums of Q are zero, $\theta = 0$ is a root of (6), say $\theta_N = 0$. We observe that both b_1 and b_2 contain the factor $1/\theta$. If all the roots of (6) are distinct, we write

$$\begin{aligned} \underline{F}(\theta) = & \left[\sum_{k=1}^{N-1} \frac{A_k}{\theta - \theta_k} + \frac{A_N}{\theta^2} + \frac{A_{N+1}}{\theta} \right] e^{-\theta B_1} \\ & + \left[\sum_{k=1}^{N-1} \frac{B_k}{\theta - \theta_k} + \frac{B_N}{\theta^2} + \frac{B_{N+1}}{\theta} \right] e^{-\theta B_2} \end{aligned} \quad (7)$$

where

$$\begin{aligned} A_k &= \lim_{\theta \rightarrow \theta_k} (\theta - \theta_k) (Q + \theta D)^{-1} b_1 \\ A_N &= \lim_{\theta \rightarrow 0} \theta^2 (Q + \theta D)^{-1} b_1 \\ A_{N+1} &= \lim_{\theta \rightarrow 0} \frac{\partial}{\partial \theta} (\theta^2 (Q + \theta D)^{-1}) b_1 \end{aligned}$$

and $\{B_k\}_{k=1}^{N+1}$ are similarly defined with b_1 replaced by b_2 . If there are repeated roots of (6), the partial fraction expansion can be obtained accordingly.

Since we are interested in the inverse Laplace transform within the region $B_1 \leq x \leq B_2$, we perform the operation for only the first term in (7). The application of the boundary conditions in obtaining b_1 and b_2 nullifies the solution outside $B_1 \leq x \leq B_2$. Let $\underline{H}(x) = [E[\tau | 1, x], \dots, E[\tau | N, x]]^T$ be the inverse Laplace transform of $\underline{F}(\theta)$. We have for $B_1 \leq x \leq B_2$

$$\begin{aligned} \underline{H}(x) = & \sum_{k=1}^{N-1} A_k e^{\theta_k(x-B_1)} + A_N \frac{(x-B_1)^2}{2} \\ & + A_{N+1}(x-B_1). \end{aligned} \quad (8)$$

The complexity of the main computations in obtaining $\underline{H}(x)$ is in the order of N^3 by the standard Gaussian elimination for the solutions of b_1 and b_2 .

After calculating the mean threshold crossing times, the relative frequency of each rate can also be obtained. Let f_l , f_h , and f_m be the proportion of time each rate is used. Obviously

$$f_l + f_h + f_m = 1. \quad (9)$$

Furthermore, the average rate, R_{av} is given by

$$R_{av} = f_l \lambda_l + f_h \lambda_h + f_m \lambda_m. \quad (10)$$

Consider a long period t . Let n_h and n_l respectively be the number of times T_l and T_h are crossed during t . Since each low or high rate period is followed by a period with medium rate $n_l(E[\tau_1] + E[\tau_3]) + n_h(E[\tau_2] + E[\tau_3]) = t$. The average rate over period t is R_{av} , i.e., $n_l(E[\tau_1]\lambda_l + E[\tau_3]\lambda_m) + n_h(E[\tau_2]\lambda_h + E[\tau_3]\lambda_m) = R_{av}t$. Combining the last two equations we have

$$\frac{n_l}{n_h} = \frac{E[\tau_2](\lambda_h - R_{av}) + E[\tau_3](\lambda_m - R_{av})}{E[\tau_1](R_{av} - \lambda_l) + E[\tau_3](R_{av} - \lambda_m)}. \quad (11)$$

On the other hand

$$\frac{f_l}{f_h} = \frac{n_l E[\tau_1]}{n_h E[\tau_2]}. \quad (12)$$

We obtain f_l , f_h , and f_m by solving (9)–(12).

The analysis technique developed in this section is applicable to any Markov chain model. In the following subsection, as an example, we derive the solutions for the mean first passage times in detail for a two-state Markov chain.

B. Two-State Markov Chain Model

We model the process describing the rate at which the bits are removed from the receive buffer with a two-state Markov chain. For video encoding schemes that use only interframe technique without motion estimation, a two-state model can be applicable. A video sequence containing periods of high and low motion can be modeled by two states, one state with low bitrate (corresponding to low motion) and the other state with high bitrate (corresponding to high motion). Let

$$Q = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix}$$

be the state transition rate matrix of the Markov chain. The parameters $1/\alpha$ and $1/\beta$ are matched to average durations of high and low motion periods. Without loss of generality assume that $R_1 < R_2$. The roots $\theta_{1,2}$ of (6) are given by

$$\theta_1(\lambda) = \frac{\beta(\lambda - R_1) + \alpha(\lambda - R_2)}{(\lambda - R_1)(\lambda - R_2)}, \quad \theta_2(\lambda) = 0.$$

The high rate λ_h is chosen to be larger than both R_1 and R_2 in order to have a drift toward T_m after hitting T_l , whereas the low rate λ_l is similarly selected smaller than both R_1 and R_2 . The medium rate λ_m is chosen as the average rate, $\lambda_m = (R_1\beta + R_2\alpha)/(\alpha + \beta)$. Hence, we have

$$\lambda_l < R_1 < \lambda_m < R_2 < \lambda_h.$$

We find the mean values of τ_1 , τ_2 , and τ_3 separately.

1) $E[\tau_1]$ (mean first passage time from L to M): Since $R_1 < \lambda_m < R_2$, the chain is in state 1 when T_h is hit, i.e., $E[\tau_1] = E[\tau_1 | 1]$. By doing the partial fraction expansion in (7) and taking the inverse Laplace transform with respect to θ , with $\text{Re}\{\theta\} > 0$, we obtain

$$\begin{aligned} E[\tau_1] = & -\frac{(\alpha + \beta)(T_h - T_m)}{\beta(\lambda_l - R_1) + \alpha(\lambda_l - R_2)} \\ & + \frac{\alpha(R_1 - R_2)(\lambda_l - R_2)}{[\beta(\lambda_l - R_1) + \alpha(\lambda_l - R_2)]^2} (1 - e^{\theta_1(\lambda_l)(T_h - T_m)}). \end{aligned} \quad (13)$$

The first term in (13) corresponds to the steady-state behavior of the Markov chain, whereas the second term reflects the effect of the initial state.

2) $E[\tau_2]$ (mean first passage time from H to M): Since $R_1 < \lambda_m < R_2$, the chain is in state 2 when T_1 is hit, i.e., $E[\tau_2] = E[\tau_2 | 2]$. By doing the partial fraction expansion in (7) and taking the inverse Laplace transform with respect to θ , with $\text{Re}\{\theta\} < 0$, we obtain

$$E[\tau_2] = \frac{(\alpha + \beta)(T_m - T_1)}{\beta(\lambda_h - R_1) + \alpha(\lambda_h - R_2)} + \frac{\beta(R_2 - R_1)(\lambda_h - R_1)}{[\beta(\lambda_h - R_1) + \alpha(\lambda_h - R_2)]^2} (1 - e^{\theta_1(\lambda_h)(T_1 - T_m)}) \quad (14)$$

3) $E[\tau_3]$ (mean first passage time from M to either L or H): The state of the chain at the beginning can be either one or two. Since $R_1 < \lambda_m < R_2$, we need to evaluate the boundary conditions. Solving the linear system of equations we obtain the two unknown constants:

$$\gamma_1 = \frac{(\alpha + \beta)(T_h - T_1)^2/2 - (\lambda_m - R_2)(T_h - T_1)}{\beta(T_h - T_1) - (\lambda_m - R_2)}$$

$$\gamma_2 = \frac{(\alpha + \beta)(T_h - T_1)^2/2 - (\lambda_m - R_1)(T_h - T_1)}{\lambda_m - R_1 - \alpha(T_h - T_1)}$$

Performing the partial fraction expansion and taking the inverse Laplace transform we obtain (15) and (16), see equations at the bottom of this page.

Example: We choose the following parameters for the two-state Markov chain

$$\alpha = \beta = 0.2 \text{ s}^{-1}$$

and

$$R_1 = 0.7 \text{ Mb/s}, R_2 = 1.3 \text{ Mb/s}.$$

These parameters correspond to a video sequence containing low and high motion of equal amount and each low or high period last for 5 s on the average. The average rate is 1.0 Mb/s. We choose the following transmission rates

$$\lambda_l = 0.5 \text{ Mb/s}, \lambda_m = 1.0 \text{ Mb/s}, \lambda_h = 1.5 \text{ Mb/s}.$$

The thresholds T_l , T_m , and T_h are chosen as

$$T_l = 0.1B_M, T_m = 0.5B_M \text{ and } T_h = 0.9B_M$$

and B_M varies from 1 Mb/s to 10 Mb/s (1 s to 10 s of video). The mean first passage times are shown in Fig. 10 as a function

of buffer size, where the solid curve represents $E[\tau_1] = E[\tau_2]$ and the dashed curve corresponds to $E[\tau_3]$. Although the parameters used in the plots in Fig. 10 are not matched to the specific rate process used for the experimentation in Section IV, the curves show similar trends with the curves in Fig. 7.

When B_M is large, $E[\tau_1]$ and $E[\tau_2]$ are dominated by the first term in (13) and (14) respectively. The reason for this is that as B_M gets larger, the mean first passage times also increase and the effect of the initial state is washed out.

When the transmission rate is λ_m , the average drift is zero. $E[\tau_3]$ is determined by the occurrence of a temporary overload or underload period whose duration should increase as B_M gets larger. Since the time spent at each state is exponentially distributed, the probability of a long overload or underload period decreases exponentially as B_M increases. Therefore, $E[\tau_3]$ increases exponentially with the buffer size. For large buffer size, starting at the medium threshold, the mean time to hit the lower or upper threshold is large compared to mean passage times from lower and upper thresholds to the medium threshold. Therefore, f_m is much larger than f_l and f_h when the receive buffer is large (at $B_M = 10 \text{ Mb}$, $f_m = 5/6$ and $f_l = f_h = 1/12$).

VI. CONCLUSION

We have demonstrated the adaptiveness and efficiency of client-based rate and synchronization control through the GRAMS implementation. Currently, we are refining the interface and performing further traffic studies based on real applications. Also, we presented new performance analysis methodologies using first passage time analysis, for which, both the state of the buffer and the state of the video encoding algorithm are considered.

The current GRAMS server as well as the client interface and control are functioning as expected in our current implementation. Further experience will be gained and reported as the server and client are deployed locally on our campus.

APPENDIX A IMPLEMENTATION OF MULTIMEDIA SERVER AND CLIENT INTERFACE

In GRAMS, three types of media are distinguished and a queue is associated with each media type. In decreasing order of service priority, they are as follows:

Continuous real-time media: This includes video (CBR and VBR video) and audio that requires continuous transmission and playback. The major QoS concerns are latency and jitter as defined earlier.

$$E[\tau_3 | 1] = \frac{(\lambda_m - R_2)\gamma_1 - (\beta\gamma_1 + \lambda_m - R_2)(T_m - T_1) + (\alpha + \beta)(T_m - T_1)^2/2}{(\lambda_m - R_1)(\lambda_m - R_2)} \quad (15)$$

$$E[\tau_3 | 2] = \frac{(\lambda_m - R_1)\gamma_2 + (\lambda_m - R_1 - \alpha\gamma_2)(T_m - T_1) - (\alpha + \beta)(T_m - T_1)^2/2}{(\lambda_m - R_1)(\lambda_m - R_2)} \quad (16)$$

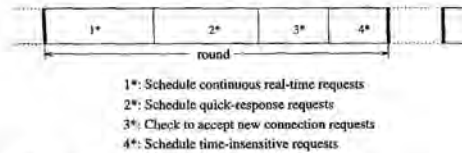


Fig. 11. Basic frame structure.

Quick response data: This includes images, postscript files for transparencies, etc., for which the major QoS concern is latency.

Time insensitive data: This includes all sorts of retrieval data that do not have strict delay requirements.

The service of these three queues is done in a round robin fashion in rounds of fixed duration (Fig. 11). A packet is sent for each connection in a round. The rate of the connection is therefore given by the packet size divided by the duration of the round. Rate assignment is done by packet size assignment during connection setup. Rate adjustment by changing the packet size occurs when feedback requests are made downstream by the client. The packets are served according to the priority of the media type described above. Transmission starvation (or jitter) occurs for a connection when a packet for that connection is not transmitted within that round. Cache starvation occurs when a packet cannot be transmitted in a round due to cache underflow.

We monitor the QoS at the server by computing the underflow jitter rate of the transmit value. The starvation count is updated per round for each connection i according to

$$C_i \leftarrow \alpha_i C_i + I_i \quad (17)$$

in which $I_i = 1$ if, during the round no packet is sent due to cache underflow or transmission starvation. Compensation of transmission may occur if the round has time left over and if C_i is positive. In this case, I_i is equal to one minus the number of packets transmitted for the connection i . For quick response data services such as image transport, we set $\alpha_i = 1$ so that C_i represents the cumulative transmission deficit. For video transmission, we set $\alpha_i < 1$ so that C_i represents a geometrically weighted running sum of the transmission deficit.

The single parameter C_i keeps track of the QoS, for which, the server is trying to maintain for connection i . However, the actual QoS perceived by the client is better than C_i because further buffering and smoothing of jitter is performed at the client. Nevertheless, the local server QoS parameter is useful because the degradation of this QoS parameter can result from usage congestion of any resource, such as; the disk bus, cache memory, transmission bandwidth, or processing cycles. When the parameter exceeds a certain threshold, GRAMS attempts to reduce usage congestion by exercising rate control and admission control. Rate control reduces the rates of quick response data by actions such as reducing the packet size for images. Admission control delays new admissions until the QoS parameter recedes. Fig. 12 plots the overall system starvation rate as a function of the offered load, α , and thresholds for video. As expected, starvation

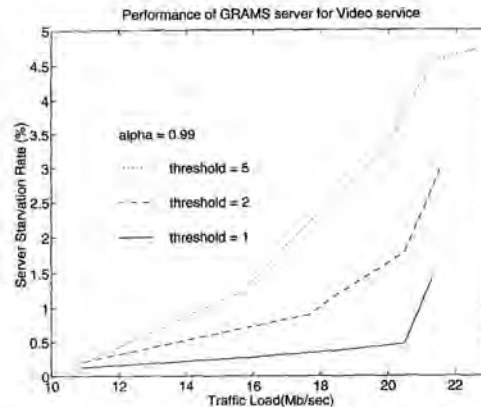


Fig. 12. Server starvation rate.

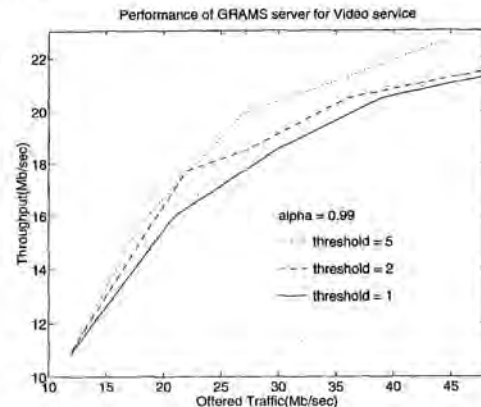


Fig. 13. Server throughput.

increases as offered load is increased. Starvation can occur even at low load because of the rather unpredictable load in each round for the operating system, making available processing cycles in each round for caching and transmission functions somewhat unpredictable. Such starvation has the effect of gating admission very much the same way as a traffic light. Fig. 13 plots the carried load as a function of the offered load (assuming failed admission are cleared instead of being queued). The difference between carried load and the offered load reflects the fraction of connection requests that suffers delay (often minimal) before the connection is completed. Thus, we have a system that is adaptive and degrades gracefully as system load is increased, and requires very little knowledge of traffic statistics.

APPENDIX B

In this Appendix, we derive the Laplace transform $F_s(\theta)$ of $E[\tau | s, x]$ with respect to x and show that the N -dimensional vector $F(\theta)$ given by (4) can be represented by (5).

Case 1: $\lambda > R_s$

Taking the Laplace transform of (1), we have

$$\begin{aligned} F_s(\theta) &= \int_{B_1}^{B_2} e^{-\theta x} e^{-q_s \frac{B_2-x}{\lambda-R_s}} \frac{B_2-x}{\lambda-R_s} dx + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x} \\ &\quad \cdot \int_0^{\frac{B_2-x}{\lambda-R_s}} e^{-q_s t} [t + E[\tau | s', x + (\lambda - R_s)t]] dt dx \\ &= \frac{1}{q_s} \int_{B_1}^{B_2} e^{-\theta x} \left(1 - e^{-q_s \frac{B_2-x}{\lambda-R_s}} \right) dx \\ &\quad + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x'} E[\tau | s', x'] \\ &\quad \cdot \int_0^{\frac{x'-B_1}{\lambda-R_s}} e^{-(q_s - \theta(\lambda-R_s))t} dt dx' \end{aligned}$$

where we used the change of variable $x' = x + (\lambda - R_s)t$ to obtain the second term.

$$\begin{aligned} F_s(\theta) &= \frac{1}{q_s} \left[\frac{1}{\theta} (e^{-\theta B_1} - e^{-\theta B_2}) + \frac{(\lambda - R_s) e^{-\frac{\theta q_s B_2}{\lambda-R_s}}}{q_s - \theta(\lambda - R_s)} \right. \\ &\quad \cdot \left(e^{-(\theta - \frac{q_s}{\lambda-R_s})B_1} - e^{-(\theta - \frac{q_s}{\lambda-R_s})B_2} \right) \Big] \\ &\quad + \frac{1}{q_s - \theta(\lambda - R_s)} \sum_{s' \neq s} q_{ss'} \\ &\quad \cdot \left[F_{s'}(\theta) - e^{-(\theta - \frac{q_s}{\lambda-R_s})B_1} \right. \\ &\quad \cdot \left. \int_{B_1}^{B_2} E[\tau | s', x'] e^{-\frac{\theta q_s}{\lambda-R_s} x'} dx' \right] \\ &= \frac{1}{\theta q_s (q_s - \theta(\lambda - R_s))} \left[e^{-\theta B_1} (q_s - \theta(\lambda - R_s)) \right. \\ &\quad \cdot (1 - e^{-\frac{q_s}{\lambda-R_s}(B_2-B_1)}) - e^{-\theta B_2} q_s \Big] \\ &\quad + \frac{1}{q_s - \theta(\lambda - R_s)} \sum_{s' \neq s} q_{ss'} \\ &\quad \cdot \left[F_{s'}(\theta) - e^{-(\theta - \frac{q_s}{\lambda-R_s})B_1} F_{s'}\left(\frac{q_s}{\lambda - R_s}\right) \right]. \quad (18) \end{aligned}$$

Case 2: $\lambda < R_s$

Taking the Laplace transform of (2), we have

$$\begin{aligned} F_s(\theta) &= \int_{B_1}^{B_2} e^{-\theta x} e^{-q_s \frac{x-B_1}{\lambda-R_s}} \frac{x-B_1}{\lambda-R_s} dx \\ &\quad + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x} \int_0^{\frac{x-B_1}{\lambda-R_s}} e^{-q_s t} \\ &\quad \cdot [t + E[\tau | s', x + (\lambda - R_s)t]] dt dx \\ &= \frac{1}{q_s} \int_{B_1}^{B_2} e^{-\theta x} \left(1 - e^{-q_s \frac{x-B_1}{\lambda-R_s}} \right) dx \\ &\quad + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x'} E[\tau | s', x'] \\ &\quad \cdot \int_0^{\frac{B_2-x'}{\lambda-R_s}} e^{-(q_s - \theta(\lambda-R_s))t} dt dx' \end{aligned}$$

where we used the change of variable $x' = x + (\lambda - R_s)t$ to obtain the second term. Evaluating the above integrals we have

$$\begin{aligned} F_s(\theta) &= \frac{1}{\theta q_s (q_s - \theta(\lambda - R_s))} \left[e^{-\theta B_1} q_s - e^{-\theta B_2} \right. \\ &\quad \cdot \left(q_s - \theta(\lambda - R_s) \left(1 - e^{-\frac{q_s}{\lambda-R_s}(B_2-B_1)} \right) \right) \Big] \\ &\quad + \frac{1}{q_s - \theta(\lambda - R_s)} \sum_{s' \neq s} q_{ss'} \\ &\quad \cdot \left[F_{s'}(\theta) - e^{-(\theta - \frac{q_s}{\lambda-R_s})B_1} F_{s'}\left(\frac{q_s}{\lambda - R_s}\right) \right]. \quad (19) \end{aligned}$$

Comparing (18) and (19) we observe that both equations have the same form with B_1 and B_2 interchanged. Multiplying both sides of (18) and (19) with $q_s - \theta(\lambda - R_s)$ and arranging the terms, we obtain the matrix equation

$$(Q + \theta D) \underline{F}(\theta) = \underline{b}(\theta)$$

where $\underline{b}(\theta) = [b_1(\theta), \dots, b_N(\theta)]^T$ is given by

$$\begin{aligned} b_s(\theta) &= \frac{1}{\theta q_s} \left[-e^{-\theta B_1} (q_s - \theta(\lambda - R_s)) \right. \\ &\quad \cdot \left(1 - e^{-\frac{q_s}{\lambda-R_s}(B_2-B_1)} \right) + e^{-\theta B_2} q_s \Big] \\ &\quad + e^{-(\theta - \frac{q_s}{\lambda-R_s})B_1} \sum_{s' \neq s} q_{ss'} F_{s'}\left(\frac{q_s}{\lambda - R_s}\right) \quad (20) \end{aligned}$$

for a state s with $\lambda > R_s$. For s with $\lambda < R_s$, $b_s(\theta)$ is the same as (20) with B_1 and B_2 interchanged. $b_s(\theta)$ contains $N(N-1)$ unknown constants $\{F_{s'}(\frac{q_s}{\lambda-R_s})\}_{s' \neq s}$.

These constants correspond to the discontinuity of $E[\tau | s, x]$ at B_1 or B_2 . For a state s with $\lambda > R_s$

$$\lim_{x \nearrow B_2} E[\tau | s, x] = 0$$

whereas

$$\lim_{x \searrow B_1} E[\tau | s, x] > 0$$

although $E[\tau | s, B_1] = 0$. Similarly, for a state s with $\lambda < R_s$, the discontinuity occurs at B_2 . We can write for s with $\lambda > R_s$

$$b_s(\theta) = -\frac{1}{\theta} (e^{-\theta B_1} - e^{-\theta B_2}) + \gamma_s e^{-\theta B_1} \quad (21)$$

and for a state s with $\lambda < R_s$

$$b_s(\theta) = -\frac{1}{\theta} (e^{-\theta B_1} - e^{-\theta B_2}) + \gamma_s e^{-\theta B_2} \quad (22)$$

where the unknown constants $\{\gamma_s\}$ can be expressed from (20). These constants are evaluated by taking the inverse Laplace transform of $\underline{F}(\theta) = (Q + \theta D)^{-1} \underline{b}(\theta)$ and applying the following boundary conditions: For each state s with $\lambda > R_s$, we set $E[\tau | s, B_2] = 0$ and for s with $\lambda < R_s$, we set $E[\tau | s, B_1] = 0$. Since $\{\gamma_s\}$ appears as a coefficient in the corresponding mean passage time equations, $\{\gamma_s\}$ can be obtained as the solution of a system of N linear equations.

REFERENCES

- [1] T. D. C. Little and F. Kao, "An intermedia skew control system for multimedia data presentation," in *Proc. 3rd Int. Workshop Network Operating Syst. Support Digital Audio Video*, La Jolla, CA, Nov. 1992.
- [2] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, Apr. 1990.
- [3] —, "Multimedia synchronization protocols for broadband integrated services," *IEEE J. Select. Areas Commun.*, vol. 9, no. 9, Dec. 1991.
- [4] N. U. Qazi, M. Woo, and A. Ghafoor, "A synchronization and communication model for distributed multimedia objects," in *Proc. ACM Multimedia*, Anaheim, CA, 1993.
- [5] P. V. Rangan, S. Ramanathan, and H. M. Vin, "Techniques for multimedia synchronization in network file systems," *IEEE Comp. Commun.*, vol. 16, no. 3, Mar. 1993.
- [6] P. V. Rangan and H. M. Vin, "Designing an on-demand multimedia service," *IEEE Commun. Mag.*, vol. 30, no. 7, July 1992.
- [7] H. P. Katsch and B. S. Robinson, "Predictive pre-fetch in the nemesis multimedia information service," in *Proc. ACM Multimedia*, San Francisco, Oct. 1994.
- [8] P. Hoepner, "Synchronizing the presentation of multimedia objects," *IEEE Commun. Mag.*, vol. 15, no. 9, Nov. 1992.
- [9] R. Steinmetz, "Synchronization properties in multimedia systems," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, Apr. 1990.
- [10] D. P. Anderson and G. Homsy, "A continuous media I/O server and its synchronization mechanism," *IEEE Comp.*, vol. 24, no. 10, Oct. 1991.
- [11] D. Shepherd, D. Hutchinson, F. Garcia, and G. Coulson, "Protocol support for distributed multimedia applications," *Comp. Commun.*, vol. 13, no. 7, Sept. 1990.
- [12] J. Y. Hui, J. Zhang, and J. Li, "Quality of service control in GRAMS for ATM local area network," *IEEE J. Select. Areas Commun.*, vol. 13, no. 4, May 1995.
- [13] D. Minoli and R. Keinath, *Distributed Multimedia Through Broadband Communications Services*. Norwood, MA: Artech House, 1994.
- [14] J. Y. Hui, "Resource allocation for broadband networks," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, Dec. 1988.
- [15] A. Pitsillides and J. Lambert, "Adaptive connection admission and flow control: Quality of service with high utilization," in *Proc. Conf. Comp. Commun. (IEEE Infocom)*, Toronto, Canada, June 1994.
- [16] G. Ramamurthy and R. S. Dighe, "Distributed source control: a network access control for integrated broadband packet networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 990–1002, Sept. 1991.
- [17] K. Obraczka *et al.*, "Internet resource discovery services," *IEEE Comput.*, pp. 8–22, Sept. 1993.
- [18] G. Branwyn, *Mosaic Quick Tour: Accessing & Navigating the Internet's World Wide Web*. Chapel Hill, NC: Ventana Press, 1994.
- [19] SunSoft Incorporated, "CellB Codec," in *Imaging Library Programmer's Guide (Solaris X11 1.1 Imaging Library AnswerBook)*.
- [20] W. Stallings, *Advances in ISDN and Broadband ISDN*. Los Alamitos, CA: IEEE Computer Society Press, 1992.
- [21] D. Reininger, B. Melamed, and D. Raychaudhuri, "Variable bitrate MPEG video: characteristics, modeling and multiplexing," in *Proc. 14th Int. Teletraffic Congress*, Antibes Juan-les-Pins, France, June 6–10, 1994.
- [22] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of data handling systems with multiple sources," *Bell Syst. Tech. J.*, vol. 61, no. 8, Oct. 1982.



Joseph Y. Hui (S'82-M'83-SM'90) received the S.B., S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1981, 1981, 1982, and 1983, respectively.

He joined Bell Communications Research, Morristown, NJ, in 1983, and Rutgers University, New Brunswick, NJ, in 1989 as an Associate Professor in the Department of Electrical and Computer Engineering. From 1979 to 1981, he worked at the Comsat Laboratories under the MIT Cooperative

Program. He was also an Adjunct Professor at the Center for Telecommunications Research, Columbia University, New York, NY, from 1987 to 1988. His current research interests include coding and information theory, wireless information networks, and switching, protocol, traffic, and security issues for integrated broadband networks. He is the author of a book entitled *Switching and Traffic Theory for Integrated Broadband Networks* (Kluwer Academic, 1990).

Dr. Hui is the recipient of the 1985 William Bennett Prize Paper Award given by the IEEE Communications Society. His work on multiple access communications and switching networks resulted in several patents. He also received the Presidential Young Investigator Award from the National Science Foundation.



Ezhan Karasan (S'89) received the B.S. degree in electrical engineering from the Middle East Technical University, Ankara, Turkey and the M.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 1987 and 1990, respectively. He is currently pursuing his Ph.D. degree in electrical engineering at Rutgers University, New Brunswick, New Jersey.

His research interests include performance analysis of broadband networks and multimedia systems.

Mr. Karasan was awarded a fellowship from the NATO Science Scholarship Program from 1991 to 1994.



Jun Li (S'93) received the B.Sc. and M.Sc. degrees in electrical engineering from Xidian University, Xi'an, PRC, in 1983 and 1985, respectively. He is currently a Ph.D. student in the ECE department of Rutgers University, specializing in wireless multimedia services and broadband networks.

He worked as an assistant professor at Beijing Information Technology Institute, Beijing, PRC, from 1986 to 1990, working on speech synthesis and recognition. He served as a visiting researcher at NEC's C&C Central Labs, Kawasaki, Japan, from 1987 to 1989. From 1990 to 1993, he worked for Data Control Limited, Tokyo, Japan and San Francisco, CA, as a software engineer, developing LAN softwares for multi-platform computers.



Junbiao Zhang received the B.S. degree in computer science from the University of Science and Technology of China, in 1990, and the M.S. degree in computer science from Rutgers University, Piscataway, NJ, in 1994. He is now a Ph.D. student in the computer science department at Rutgers University. His research interests include distributed systems, high speed networks, and multimedia systems. He also designed and implemented the multimedia service system GRAMS.

Ex. 1019A

IEEE
JOURNAL ON
SELECTED
AREAS IN
COMMUNI-
CATIONS

14
JAN-JUNE
1996

TK 51Ø1
.A1 I35



IEEE JOURNAL ON

SELECTED AREAS IN COMMUNICATIONS

A PUBLICATION OF THE IEEE COMMUNICATIONS SOCIETY



JANUARY 1996

VOLUME 14

NUMBER 1

ISACEM

(ISSN 0733-8716)

SYNCHRONIZATION ISSUES IN MULTIMEDIA COMMUNICATIONS

Guest Editors—N. D. Georganas, R. Steinmetz, and T. Nakagawa

Guest Editorial	N. D. Georganas, R. Steinmetz, and T. Nakagawa	1
-----------------------	--	---

PAPERS

Overview and Reference Models

A Media Synchronization Survey: Reference Model, Specification, and Case Studies ...	G. Blakowski and R. Steinmetz	5
A Temporal Reference Framework for Multimedia Synchronization	M. J. Pérez-Luque and T. D. C. Little	36
Continuity and Synchronization in MPEG	P. V. Rangan, S. S. Kumar, and S. Rajan	52

User Perception

Human Perception of Jitter and Media Synchronization	R. Steinmetz	61
--	--------------	----

Modeling

A New Multimedia Synchronization Model	M. Haindl	73
Modeling Logical and Temporal Synchronization in Hypermedia Systems	P. Sénac, M. Díaz, A. Léger, and P. de Saqui-Sannes	84
Synchronization Representation and Traffic Source Modeling in Orchestrated Presentation	S. V. Raghavan, B. Prabhakaran, and S. K. Tripathi	104
Doing FLIPS: FLEXible Interactive Presentation Synchronization	J. Schnepf, J. A. Konstan, and D. H.-C. Du	114
Using Timed CSP for Specification Verification and Simulation of Multimedia Synchronization	A. F. Ates, M. Bilgic, S. Saito, and B. Sarikaya	126
An EFSM-Based Multimedia Synchronization Model and the Authoring System	C.-M. Huang and C.-M. Lo	138

Mechanisms

Distributed Source-Destination Synchronization Using Inband Clock Distribution	C.-S. Li and Y. Ofek	153
Multimedia Group Synchronization Protocols for Integrated Services Networks	I. F. Akyildiz and W. Yen	162
Clock Hierarchies: An Abstraction for Grouping and Controlling Media Streams	K. Rothermel and T. Helbig	174
A General-Purpose Multimedia Synchronization Mechanism Based on Causal Relations	J.-P. Courtiat, L. Fernando Rust da Costa Carmo, and R. Cruz de Oliveira	185

Fault Recovery

Exception Handling and Fault-Tolerance in Multimedia Synchronization	K. Naik	196
--	---------	-----

(Continued on Back Cover)

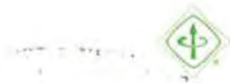
(Continued from Front Cover)

<i>Implementation</i>	
A Multimedia Synchronization Model and Its Implementation in Transport Protocols	<i>C.-C. Yang and J.-H. Huang</i> 212
Client-Server Synchronization and Buffering for Variable Rate Multimedia Retrievals	<i>J. Y. Hui, E. Karasan, J. Li, and J. Zhang</i> 226
MultiSync: A Synchronization Model for Multimedia Systems	<i>H.-Y. Chen and J.-L. Wu</i> 238
A Programming Model and System Infrastructure for Real-Time Synchronization in Distributed Multimedia Systems ..	<i>G. S. Blair, G. Coulson, M. Papathomas, P. Robin, J.-B. Stefani, F. Horn, and L. Hazard</i> 249
Synchronization of Multimedia Data for a Multimedia News-on-Demand Application	<i>L. Lamont, L. Li, R. Brimont, and N. D. Georganas</i> 264
CALL FOR PAPERS	
Personal Communications—Services, Architecture and Performance Issues	279
Networking and Performance Issues of Personal Mobile Communications	280
Real-Time Video Services in Multimedia Networks	281
Advances in ATM Switching Systems for B-ISDN	282
Computer-Aided Modeling, Analysis and Design of Communications	283
IEEE COPYRIGHT FORM	284

Upcoming Issues of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS

Topic	Date of Call*
Circuit Switching for Broadband ISDN and Beyond	June 1994
Wireless Local Communications	June 1994
Optical Networks	December 1994
Integrated Video-Based Services for the Home	December 1994
Distributed Multimedia Systems and Technology	December 1994
Code Division Multiple Access Networks III and IV	January 1995
Wireless ATM	May 1995
Computational and Artificial Intelligence in High Speed Networks	August 1995
Network Support for Multipoint Communication	August 1995
Advances in ATM Switching Systems for B-ISDN	December 1995
Real-Time Video Services in Multimedia Networks	December 1995
Computer-Aided Modeling, Analysis, and Design of Communication Links	December 1995
Personal Communications—Services, Architecture, and Performance Issues	December 1995
Networking and Performance Issues of Personal Mobile Communications	December 1995

*Issue of J-SAC in which the Call for Papers first appears.



IEEE COMMUNICATIONS SOCIETY

The field of interest of the IEEE Communications Society consists of all telecommunications including telephone, telegraphy, facsimile, and point-to-point television, by electromagnetic propagation including radio; wire, aerial, underground, coaxial, and submarine cables; waveguides, communication satellites, and lasers; in marine, aeronautical, space, and fixed station services; repeaters, radio relaying, signal storage, and regeneration; telecommunication error detection and correction; multiplexing and carrier techniques; communication switching systems; data communications; and communication theory.

In addition to the above, this JOURNAL or the IEEE TRANSACTIONS ON COMMUNICATIONS contains papers pertaining to analog and digital signal processing and modulation, audio and video encoding techniques, the theory and design of transmitters, receivers, and repeaters for communications via optical and sonic media, the design and analysis of computer communication systems, and the development of communication software. Contributions of theory enhancing the understanding of communication systems and techniques are included, as are discussions of the social implications of the development of communication technology. All members of the IEEE are eligible for membership in the Society upon payment of the annual Society membership fee of \$23.00. Members may receive this JOURNAL upon payment of an additional \$25.00 (\$48.00 total), the IEEE TRANSACTIONS ON COMMUNICATIONS upon payment of an additional \$27.00 (\$50.00 total), or both publications upon payment of an additional \$52.00 (\$75.00 total). For information on joining, write to the IEEE at the address below. Member copies of Transactions/Journals are for personal use only.

IEEE COMMUNICATIONS SOCIETY JOURNAL Editorial Board 1996

T. J. PLEVYAK, *Director of Publications*
Bell Atlantic

W. H. TRANTER, *Editor-in-Chief*
Dept. Elec. Eng.
Univ. Missouri-Rolla
Rolla, MO 65401-0249
tranter@ee.umar.edu

SUE L. McDONALD, *Executive Editor*
Bellcore, Rm. 2P-291
445 South Street
Morristown, NJ 07960-1910
sue@thumper.bellcore.com

Senior Editors

A. M. BUSH
NSF
4201 Wilson Boulevard
Room 1175
Arlington, VA 22230

M. KAWASHIMA
Fujitsu Inst. Comp. Sci.
1-17-25 Shinkamata
Ota-Ku
Tokyo 144-00, Japan

T. S. RAPPAPORT
The Bradley Dept. Elec. Eng.
615 Whittemore Hall
Virginia Polytech. Inst. & State Univ.
Blacksburg, VA 24061-0111

N. K. CHEUNG
Bellcore, Rm. NVC 3Z-219
331 Newman Springs Rd.
Red Bank, NJ 07701

N. MAXEMCHUK
AT&T Bell Labs.
600 Mountain Ave.
Murray Hill, NJ 07974

W. D. SINCOSE
Bellcore, Rm. 2Q-286
445 South Street
Morristown, NJ 07960-1910

R. YATSUBOSHI
Fujitsu Labs.
1015 Kamikodanaka Nakahara-ku
Kawasaki 211, Japan

J. F. HAYES
Dept. Elec. Eng.
Concordia Univ.
1455 De Maissonneuve
Montreal, Quebec
Canada H3G 1M8

L. B. MILSTEIN
Dept. Elect. and Comput. Eng.
Mail Code 0407
Univ. of California, SD
La Jolla, CA 92093

W. BUX
IBM Res. Lab. Zurich
Saumerstrasse 4
8803 Ruschlikon ZH
Switzerland

D. P. TAYLOR
Dept. E & E Engineering
Univ. of Canterbury
Private Bag 4800
Christchurch, New Zealand

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Officers

WALLACE S. READ, *President*
CHARLES K. ALEXANDER, *President-Elect*
TSUNEO NAKAHARA, *Secretary*
HOWARD L. WOLFMAN, *Treasurer*
JERRY R. YEARGAN, *Vice President, Educational Activities*

JOEL B. SNYDER, *Vice President, Professional Activities*
W. KENNETH DAWSON, *Vice President, Publication Activities*
RAYMOND D. FINDLEY, *Vice President, Regional Activities*
DONALD C. LOUGHRY, *Vice President, Standards Activities*
BRUCE A. EISENSTEIN, *Vice President, Technical Activities*

THEODORE W. HISSEY, JR., *Executive Director*
CELIA L. DESMOND, *Director, Division III-Communications Technology Division*

Executive Staff

DANIEL J. SENESE, *General Manager*
RICHARD D. SCHWARTZ, *Staff Executive, Business Administration*
ANTHONY J. FERRARO, *Staff Executive, Publications*

IRVING ENGELSON, *Managing Director, Corporate Activities*
PETER A. LEWIS, *Managing Director, Educational Activities*
CECELIA JANKOWSKI, *Managing Director, Regional Activities*

ANDREW G. SALEM, *Managing Director, Standards Activities*
W. THOMAS SUTTLE, *Managing Director, Professional Activities*
ROBERT T. WANGEMANN, *Managing Director, Technical Activities*

Transactions/Journals Department

Staff Director: PATRICIA WALKER
Transactions Manager: GAIL S. FERENC
Electronic Publishing Manager: JERI L. UZZO
Editorial Manager: VALERIE CAMMARATA
Managing Editor: GERALDINE E. KROLIN
Senior Editor: SCOTT DISANNO
Associate Editor: LAURA L. TUCKER
Assistant Editor: LYLE SMITH

IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS is published nine times a year in January, February, April, May, June, August, September, October, and December by the Institute of Electrical and Electronics Engineers, Inc. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society/Council, or its members. **IEEE Corporate Office:** 345 East 47 Street, New York, NY 10017-2394. **IEEE Operations Center:** 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. **NJ Telephone:** 908-981-4060. **Price/Publication Information:** Individual copies: IEEE Members \$10.00 (first copy only), nonmembers \$20.00 per copy. (Note: Add \$4.00 postage and handling charge to any order from \$1.00 to \$50.00, including prepaid orders.) Member and nonmember subscription prices available upon request. Available in microfiche and microfilm. **Copyright and Reprint Permissions:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For all other copying, reprint, or republication permission, write to Copyrights and Permissions Department, IEEE Publications Administration, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. Copyright © 1996 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Second-class postage paid at New York, NY and at additional mailing offices. **Postmaster:** Send address changes to IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. GST Registration No. 125634188. Printed in U.S.A.

Guest Editorial Synchronization Issues in Multimedia Communications



I. INTRODUCTION

IN GENERAL, multimedia synchronization denotes a temporal, spatial or even logical relationship between objects, data entities, or media streams. In the context of multimedia computing and communications, it has become customary to address by synchronization the temporal relationship only [1], [2]. This notion is still very broad and it captures a large variety of issues, including inter-process communication mechanisms and concurrent programming languages. Hence, in this editorial and in all contributions to this special issue, multimedia synchronization will refer to the temporal relationship among media.

As described in [3], a multimedia system is characterized by the integrated computer-controlled generation, manipulation, presentation, storage, and communication of independent discrete (i.e., time independent, like text and graphics) and continuous (i.e., time dependent, like audio and video) media data. The digital representation of any data, and the synchronization between various kinds of media data, are the key issues for this integration. Hence, multimedia synchronization is needed to ensure a temporal ordering of events in a multimedia system.

In several publications IEEE reflects the member demands for information related to multimedia: The *IEEE Computer Magazine* devoted a first issue on Multimedia Communications in October 1985. The most recent issue to focus on multimedia appeared in May 1995. In addition, the *IEEE Multimedia Magazine*, published since 1994, is attracting a large readership. JSAC had a first issue on Multimedia Communication published in April 1990. Papers from that issue are cited throughout the multimedia community frequently, e.g., see this JSAC issue. Almost all ACM and IEEE Communication and Computer Society supported workshops and conferences related to multimedia include at least one paper related to multimedia synchronization. In conjunction with the 1995 IEEE International Conference on Multimedia Computing and Systems, T. D. C. Little and the first two guest editors of this JSAC issue organized the first and very successful workshop on multimedia synchronization.

Considerable work has been performed in the synchronization research and development area. However, there exists no

common terminology and there has been a lack of comparative work. This was the motivation for the two overviews and reference models. Fundamental work for understanding the user demands is also still in its infancy. A contribution to this issue is given in the second section. With this notion of user demands, the next section of papers addresses the topic of how to model multimedia synchronization (and therefore how to define the temporal relationships). Mechanisms to perform this comprise the next section, followed by a discussion on fault tolerance issues only (i.e., what shall happen in case of failures and how to avoid them). Finally, different implementation issues are covered, concluding with fully implemented systems and the related experiences.

This issue is therefore organized to include the following six sections:

II. OVERVIEWS AND REFERENCE MODELS

The first section is comprised of papers providing the reader with an overview of various multimedia synchronization issues. For researchers and developers active in the multimedia synchronization domain, it has been difficult until now to obtain a comprehensive overview on the various requirements, models, and mechanisms already in place. In the multimedia synchronization field there were few terms established and new upcoming papers tend to introduce again a new terminology to describe their approach. The reference models as described in the first two papers provide the reader with a framework for categorizing and comparing different approaches.

Blakowski and Steinmetz, introduce a common terminology and start with a description of the requirements from the user perspective. Subsequently, they survey the most prominent synchronization models to describe the application needs. Hence, using these models, the specification of an interface can be derived. The layered reference model presented in this paper distinguishes various layers, among which are: 1) an intra-stream layer which applies to the temporal constraints within one data stream, 2) a stream layer which allows the application to operate on continuous media streams, as well as groups of media streams (including inter-stream synchronization), 3) the object layer which hides the differences between discrete and continuous media, and 4) the fuzzy layer which allows for the specification and use of playout templates. These templates incorporate the definition of fuzzy time constraints. The paper concludes with a survey of existing work as case studies.

With their temporal reference framework for multimedia synchronization Perez-Luque and Little present a uniform theoretical basis for synchronization and temporal specifi-

Manuscript received September 1995.

N. D. Georganas is with the University of Ottawa, Ontario, Canada, K1N 6N5.

R. Steinmetz is with the IBM European Networking Center, Heidelberg, Germany.

T. Nakagawa is with the NTT Data Communications Systems Co., Kawasaki, Japan.

Publisher Item Identifier S 0733-8716(96)00223-5.

Client–Server Synchronization and Buffering for Variable Rate Multimedia Retrievals

Joseph Y. Hui, *Senior Member, IEEE*, Ezhan Karasan, *Student Member, IEEE*,
Jun Li, *Student Member, IEEE*, and Junbiao Zhang

Abstract—We consider the use of large buffers and feedback as a mechanism to maintain loosely coupled synchronization between a multimedia server and a client. The multimedia stream is modeled as a fluid flow through rate controlled valves and buffers with multiple thresholds. These thresholds are used to control the rates upstream. The quality of service for the multimedia connection is characterized in terms of the jitter in the received media stream due to buffer underflow and overflow. This quality of service is used to exercise rate and admission control in the presence of congestion. The feedback mechanism is, implemented in GRAMS, an adaptive multimedia client-server system. Experimental statistics are gathered for the purpose of traffic engineering. We employ a fluid flow and first passage time analysis to understand the traffic process through the pipelines and the buffers and to estimate the amount of signaling required by the feedback mechanism.

I. INTRODUCTION

ISSUES OF SYNCHRONIZATION AND BUFFERING FOR MULTIMEDIA

MULTIMEDIA communications, as the name implies, involve media of different types such as video, audio, image, text, and data. These communication sessions could be further classified according to their requirements for synchronization and delay. First, real time communications, such as teleconferencing, require the strictest synchronization and the least time differences among the time instants of signal capturing, signal coding, signal buffering, signal transmission, signal reception, and signal display. The end-to-end delay should typically be less than tens of milliseconds. Second, retrieval of multimedia information requires only the display of the audio-video at the same rate as its capturing. The delay between signal retrieval and signal display could be as large as several seconds. However, larger delay may result in unsatisfactory response time for the client. Also, a large delay may require the use of a large buffer somewhere between the server and the client display. Third, image transport imposes almost no synchronization requirement among the components of the signal, since the image typically is displayed upon

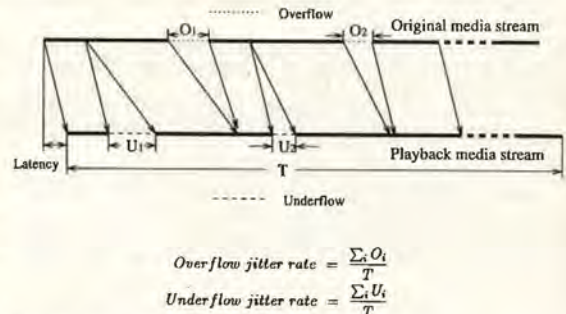


Fig. 1. Jitter rate related to underflow and overflow.

the completion of the entire image transmission. The delay of concern here is the delay between the initiation of an image request, until the complete image is received. This delay should ideally be less than a second, but delay less than 10 ms may be of no subjective advantage to the client.

We believe that delay is a significant measure of the quality of service (QoS) for multimedia. Here, we distinguish between two kinds of delay, latency versus jitter. Latency, defined for video, audio, and image transport, is the delay between signal generation (or retrieval) and signal display. Jitter, defined for continuous media streams such as audio and video, results from the disruption of the media as a consequence of underflow or overflow conditions (Fig. 1). The underflow jitter rate is defined as the fraction of time the received stream is paused due to underflowing. The overflow jitter rate is defined as the fraction of time the received stream is lost due to overflowing. The sum of these two rates is termed as the jitter rate, which measures how much jitter there is in the media stream. Jitter is due largely to the failure to synchronize the server and the client in order to maintain continuous flow. Such synchronization is achieved, in this paper, through buffer occupancy feedback.

As the current network technology supports the means to handle multimedia traffic and the user demand for multimedia services increases, protocols are required to provide satisfactory quality of service for multimedia communications. Synchronization, rate control, and buffer management between the source and the destination are three closely related problems to be solved in order to provide satisfactory quality of service for multimedia communication [1]–[11]. Different approaches have been proposed to address the problem of synchronization: models based on Petri nets [2], [4], [8],

Manuscript received August 8, 1994; revised March 30, 1995. This work was supported by the National Science Foundation under contract NSF NCR 90-58079-02 and with matching funds from Bellcore and NEC.

J. Y. Hui, E. Karasan, and J. Li are with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08855 USA.

J. Zhang is with the Department of Computer Science, Rutgers University, Piscataway, NJ 08855 USA.

Publisher Item Identifier S 0733-8716(96)00243-0.

feedback techniques for synchronizing distributed media [6], and schemes providing synchronization at the buffer level [11].

In this paper, we shall devise feedback mechanisms for the flow control and synchronization problems, analyze the stochastic behavior of the resulting problem of media data flow with feedback, and give empirical traffic data from the resulting implementation.

In particular, we use large buffers with three threshold levels to maintain loosely coupled synchronization between a multimedia server and a client. These thresholds are used to produce feedback information to exhibit rate control for preventing buffer underflow and overflow. The quality of service for the multimedia connection is expressed in terms of the jitter in the received media stream due to underflow and overflow, given in Fig. 1. This synchronization protocol is implemented in gopher-style real time ATM multimedia system (GRAMS) [12], a client-server system for multimedia communication. We characterize the resulting traffic process by using a first passage time analysis, which is used to predict the frequency of signaling required to implement the feedback mechanism.

In the multimedia server GRAMS [1], we implemented a client interface that provides client-based synchronization and QoS control via feedback. GRAMS is a distributed multimedia system that can be deployed on an ATM LAN, a wide area ATM network, or the Internet environment. The most noticeable features of this system are:

- GRAMS is fully compatible with several popular Internet document retrieval protocols including Gopher and http [17]. The client interface of GRAMS employs the popular media viewer Mosaic [18], which provides integrated multimedia display as well as the World-Wide Web (WWW) navigation help [17].
- The GRAMS server can provide heterogeneous media services such as video, audio, image, and text with QoS control.
- Several video formats (including MPEG, motion JPEG [13] and CELLB [19]) are supported in addition to image formats such as JPEG and GIF.
- GRAMS supports multiple transport protocols including TCP, UDP and ATM with AAL4 [20].

The implementation of the multimedia server in GRAMS is explained in Appendix A. Interested readers may refer to [12] for further details.

The rest of the paper is organized as follows. In Section II, we explain the feedback flow control and synchronization paradigm. The details of the flow control scheme based on three buffer thresholds are presented. In Section III, media scaling and synchronization in GRAMS are described. In Section IV, experimental results are given for the buffer occupancy and signaling frequency, obtained by using a video sequence encoded by CELLB [19]. In Section V, we analyze the traffic rate process resulting from the feedback rate control scheme. In particular, we obtain the mean time spent using a particular rate until rate changes due to feedback from the receiver end. We calculate the proportion of time each rate is used, that is used to obtain the frequency of signaling

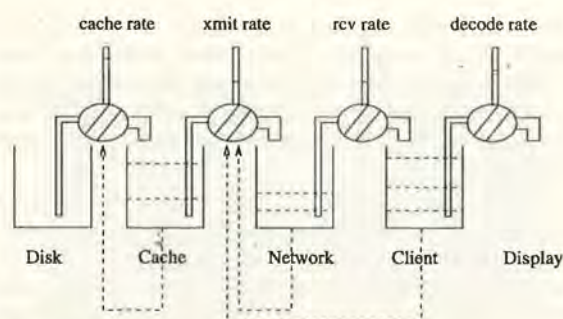


Fig. 2. Client-based rate and QoS control.

information from the client to the server. Section VI concludes the paper.

II. FEEDBACK FLOW CONTROL AND SYNCHRONIZATION PARADIGM

In the GRAMS implementation, the media stream from the server to the client may be modeled as fluid flow through a number of valves and buffers as shown in Fig. 2. There are essentially four buffers where the media stream may reside temporarily. First, the media stream may be stored in a hard disk. For teleconferencing, the media stream may be generated by the capture board, and subsequently compressed by media coding algorithms. Second, these streams are then cached at the transmit buffer. Third, the network, due to its transit latency and buffers within the network, may be viewed as another buffer. Fourth, the receive end may buffer a sufficient amount of the media stream for the purposes of maintaining a continuous stream for display and suitable synchronization with the transmit end.

Connecting these buffers are variable rate valves, some of which could be controlled using the state information of one or more of the previously described buffers. For stored video retrieval, the transfer rate from the hard disk to the system cache is dictated by the constant retrieval speed of the bus of the hard drive, even though the media stream may be of variable rate. For teleconferencing, the transfer rate could be highly variable as a function of the type of video frame (I, P and B frames for MPEG [13]) as well as the amount of motion and the granularity of the video. Some rate smoothing is necessary via the use of the transmit buffer. The transmit valve is probably the most critical control in the pipeline, performing the vital traffic shaping function. The receive valve on the client end ideally should receive information at the same rate as the transmit valve. A small transit latency may be experienced. The receive rate may match the transmit rate closely by adequate allocation of buffer and bandwidth resources [14] within the network. The video display valve empties data from the client buffer at a rate that maintains a smooth playback.

Transmitter-receiver synchronization is achieved through flow rate modulation. The control of rates as a function of buffer occupancy can be of two kinds. The first kind, that receives more attention in the literature [15], [16], consists of feedforward control, namely that the upstream buffer oc-

cupancy determines the rates of the valves downstream. For example, a high occupancy of the transmit buffer may result in a feedforward increase in the transmit rate and the display rate. A high transmit buffer occupancy could result from a surge in the media coding rate, such as, that resulting from motion in the video.

Feedforward control is often used for teleconferencing for which delay should be less than tens of milliseconds. Subsequently, the buffers in the end-to-end connection should be small. As a result, control actions may be frequent. Such frequent adjustments within the network could be difficult to achieve and often bandwidth overengineering is used to minimize the effect of rate surges. Such engineering requires certain knowledge of traffic statistics.

The focus of this paper is on the second kind of control, that consists of feedback control [5], namely the downstream buffer occupancy determines the rates of the valves. For example, a high occupancy of the receive buffer indicates that the transmit rate is too high compared to the video display bit rate. This low display rate could result from an inherently low video coding rate or the failure of the client to decode at the regular frame rate. The client then sends a feedback request to the server for a reduction in the transmit rate. Similarly, a low occupancy of the receive buffer may result in a feedback request from the client to the server for an increase in the transmit rate. A low occupancy of the receive buffer, however, can also result from a congested network, with data accumulating in the network buffer. In this circumstance, increasing the transmit rate would be detrimental. This condition should be detected by a mismatch between the requested transmit rate and the receive rate. Congestion is then alleviated by measures such as lowering the frame rate for the video and subsequently, the transmit rate.

The advantages of feedback control are many. First, this protocol emphasizes client QoS control, taking into account that the receiver has better and immediate knowledge of the condition of the end-to-end connection, including the network condition which may not be readily accessible to the transmitter. In certain sense, the effect of all upstream congestion due to the server or the network is detected, differentiated, and alleviated by the action of the client. Second, the transmitter need not examine the timing relationship and data format of the media, thus reducing the computational load of the server that often serves tens and possibly hundreds of connections at the same time. Third, the feedback automatically takes care of client-server synchronization. Fourth, this feedback mechanism does not require a detailed description of the video coding rate process other than the mean rate and hence it is robust to errors in traffic models.

If a large buffer is used, the frequency of feedback actions is low, albeit at the expense of a large delay. This infrequent change of a constant requested rate allows careful and infrequent bandwidth reservation [14]. Substantial traffic averaging is achieved as a result. Hence feedback control is more suitable for retrieval of stored media, for which a larger delay between the transmitter and the receiver is allowed.

We describe our feedback flow control and synchronization paradigm for the GRAMS system as follows. Thresholds are

set for the server cache, the network buffer, and the client buffer. For the server cache, a cache request to the hard disk is generated when half of the information in the cache has been transmitted. The cache manager then schedules a disk read to fill up the emptied half of the cache. To detect the occupancy level of the network buffer, the client keeps track of the receiver deficit as an accumulation of the difference between the requested transmit rate and the received rate. If this deficit exceeds a certain threshold, we declare the network throughput a bottleneck and actions will be taken to reduce the frame rate for the session. Thus the client interface adjusts the media quality adaptively with respect to the transport capability of the underlying network.

The receive buffer occupancy is used as control information for maintaining client-server synchronization. There are three buffer thresholds T_h , T_m , T_l , namely, the high, medium, and low thresholds. When the buffer occupancy crosses any one of these thresholds, a new transmit rate λ_h , λ_m , λ_l is requested if that rate is not the current transmission rate. The medium rate λ_m is equal to the average rate of the media, which is easily found by dividing the size of the media file by the duration of the media display. The high rate should normally be higher than the peak rate, and the low rate should normally be lower than the lowest rate. These extremal rates are more difficult to obtain than the average rate, and very often they are assigned to be a certain factor of the average rate. To safeguard against errors in estimating the peak and lowest rate, overflow or underflow conditions may be used to adjust these rates. For example, an overflow may temporarily result in a requested reduction of transmit rate to zero. Repeated overflow may necessitate a decrease of λ_l . Repeated underflow may necessitate an increase of λ_h .

In the following sections, experimental and theoretical statistics are obtained for the resulting traffic processes and buffer occupancy.

III. MEDIA SCALING AND SYNCHRONIZATION IN GRAMS

As discussed earlier, consistent underflow of the client buffer could indicate a network bottleneck or temporary congestion. In such a situation, the client requests a scale down of the media, thus reducing the bit rate requirement. In this section, we focus on media scaling and inter-media synchronization in GRAMS.

In GRAMS, a typical application is transmission of stored lectures often involving the synchronization of the video (e.g., the lecture scene), audio (e.g., the lecturer's speech that often carries most of the cognitive information of the lecture), and image streams (e.g., lecture transparencies). We call this an audio-video-visual session. The audio-video synchronization requirement is very stringent with a discrepancy of less than 100 ms, whereas, the video-image synchronization requirement could be quite loose with an allowable discrepancy in the order of seconds. Therefore, the media synchronization discussed here will primarily be between the video and audio streams.

There are two components of audio-video synchronization. First, we maintain the natural rate of playback of audio, that



Fig. 3. Structure of the video-audio block.

from a subjective point of view is more important than the natural rate of playback of video [7]. Hence, we shall maintain the continuity of the audio, at the expense of possibly introducing jitter for the video. Second, we need to synchronize the audio stream with the video stream for qualities such as lip synchronization.

Many currently available technologies record and store the audio and video streams separately. This complicates the synchronization of the two media. In the GRAMS experiments, we group the streams into audio-video groups. Each group consists of six frames of video preceded by the audio data of the group. We assume the natural frame rate of 30 frames/s. For our current implementation, we use the following three video formats: motion JPEG that consists of I frames, MPEG that consists of periodic IPP frames, and CELLB. (The MPEG standard contains B frames besides I and P frames.) The number of frames in a group was chosen so that we may restart from an I frame, if packet loss or error occur during transmission of the group, or from an I frame of the next group. Also, the grouping allows scaling back of the video rate if the network is found to be a bandwidth bottleneck (Fig. 3).

In a separate pointer file, we store headers for each audio-video group. The header makes it easy for the server to find the I frames if for some reason it needs to decimate the video stream by dropping frames. The header contains the offset of the beginning and the end of the first and third frames, which are I frames for both motion JPEG and MPEG. The audio part precedes at the location indicated by the offset of the first frame. These offsets are transmitted in the header for the group and are used both by the server and the client during media transmission and playback.

Two features complicate the design of the audio-video synchronization scheme: media scalability and transport protocol support. These two features handle problems resulting from network bandwidth bottleneck, packet loss, and server-client overload.

As discussed earlier, the client detects network congestion and requests a downward scaling of the media to relieve the bandwidth bottleneck. Our grouping would allow frame rates of 30 frames/s, 10 frames/s (by transmitting the audio plus first and third I frames only), 5 frames/s (by transmitting the audio plus the first I frame), 2.5 frames/s (by transmitting the audio for every group and the video for every other group), as well as lower rates such as 1 frames/s, every two seconds, and every five seconds. The downward adjustment is continued until the requested transmission rate matches the receive rate. Upward frame rate adjustment is initiated by the end-user through a control on the client interface. We also note that caching is performed for the maximum frame rate even though only part

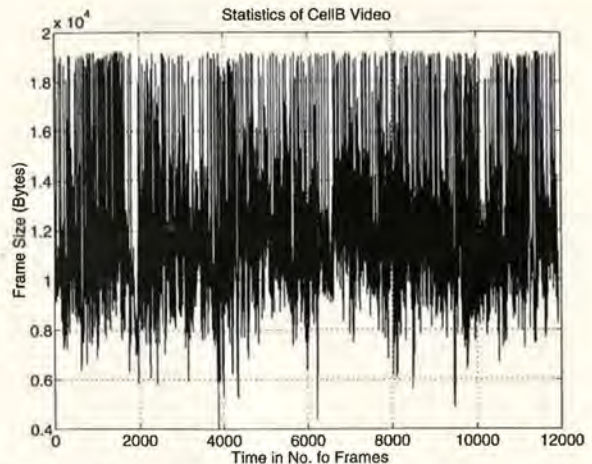


Fig. 4. A CELLB video sequence.

of the cached media may be transmitted. It is not difficult to cache only the transmitted information, but the increased disk seek time and address computations may outweigh the advantages of decreased disk reading time and reduced cache size.

This header structure devised for media scaling and synchronization is also used for accomplishing other functions of a transport protocol. The transmitter may employ a transport protocol such as TCP or UDP. Since image transport typically requires almost no packet loss and one or more seconds of image transport latency can be tolerated, using TCP as a transport mechanism could be appropriate. For audio-video applications, TCP may incur unnecessary re-transmission, delay, and processing. One option for audio-video transport is to add minimal transport layer functions within the application for packets piped via ATM and AAL 4/5 [20].

To handle out-of-order packets and packet loss, additional information is inserted into the packet header. For audio-video applications, packet loss may result in discarded or repeated audio-video segments. Currently, we do not attempt to re-transmit lost packets to avoid re-transmission delay and additional complexity in caching, re-transmission scheduling, and preventing packet duplication.

IV. EXPERIMENTAL RESULTS CONCERNING BUFFER SIZING AND RATE CONTROL

In this section, we present experimental traffic statistics as a result of client feedback based on receive buffer occupancy. The result is based on the video coding method CELLB, the currently available variable rate video encoding algorithm on our computing platform. The experiment is performed on a video stream for which a typical trajectory of the encoded size of frames is shown in Fig. 4. The video stream has the following statistics:

Number of frames $N = 12\,000$

Frame rate $R_f = 29.4$ frames/s

Bit rate $R_b = 334\,414$ bytes/s

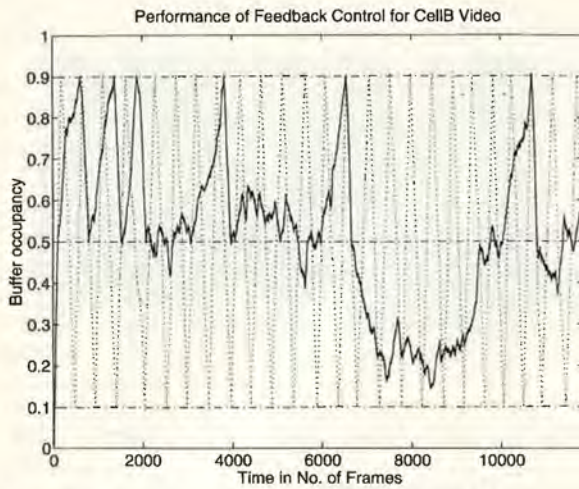


Fig. 5. Buffer occupancy variation for the CELLB video.

Statistics and modeling of variable rate MPEG can be found in [21]. Since we employ multiple audio-video standards as well as multiple sub-sampled frame rates, it seems from these rate trajectories that a proper modeling and adaptation of the model as a function of control could be quite complex. Our control approach is highly adaptive and does not require traffic statistics to be known *a-priori* other than the average rate (for the full rate and perhaps for the sub-sampled rate). Nevertheless, it is helpful to obtain traffic statistics of the requested rates by the client for the following purposes:

- Bandwidth engineering such as the sizing of bandwidth for a virtual path.
- Estimating the amount of signaling required due to these requests.
- Sizing the client buffer in order to maintain good QoS.

The choices of buffer size and thresholds affect the feedback frequency that indicates the overhead incurred by signaling and rate changes. Determining a proper combination of buffer size, thresholds and the corresponding transmission rates for a specific stream is very complicated due to many changing factors. In order to avoid overwhelming complexities, we set the three thresholds proportional to the buffer size and the three rates proportional to the average rate of the video stream. Tests are then performed to find out the buffer occupancy changes with respect to different buffer sizes. The actual values of the parameters are as follows

$$\text{Buffer size } S = T * R_b$$

$$\text{High Threshold } T_h = 0.9 * S$$

$$\text{Medium Threshold } T_m = 0.5 * S$$

$$\text{Low threshold } T_l = 0.1 * S$$

$$\text{High rate } \lambda_h = 1.5 * R_b$$

$$\text{Medium rate } \lambda_m = R_b$$

$$\text{Low rate } \lambda_l = 0.5 * R_b$$

where T ranges from 1 s to 10 s in our tests.

Fig. 5 demonstrates a typical buffer occupancy as the client consumes the video stream obtained from the GRAMS server,

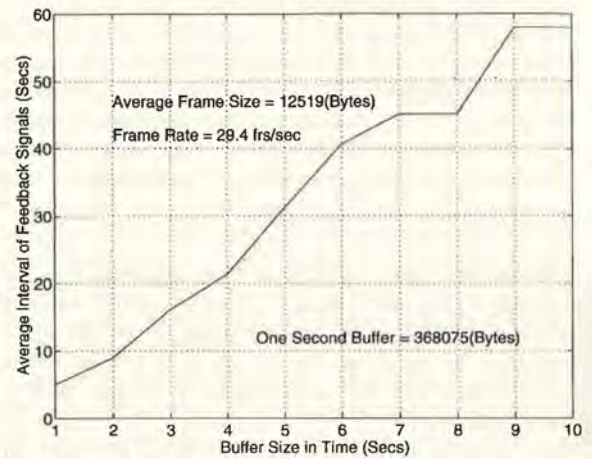


Fig. 6. Signaling frequencies relative to buffer size.

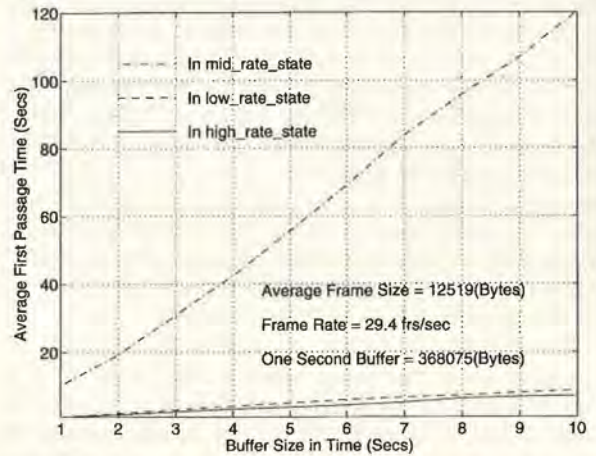


Fig. 7. First passage times relative to buffer size.

with $T = 5$ s. The curve in solid line shows the occupancy variations based on the feedback mechanism described earlier. The curve in dashed line gives the result when the medium threshold T_m is not used (i.e., the server only transmits at either the high rate λ_h or the low rate λ_l in response to client signals). In both cases, no buffer underflow or overflow occurs. We observe that the high threshold and low threshold are reached much less frequently when the medium threshold is set for the buffer. This will lead to a lower probability of buffer underflowing and overflowing. Rate changes are also reduced in this case.

Fig. 6 shows the change of average time interval between two consecutive feedback signals as the buffer size increases. The buffer size is given in the units of buffer space to store 1 s of video stream, called one second buffer (in this experiment one second buffer is approximately 368 kbytes). Even if we only use a buffer size corresponding to 1 s video stream, the average interval between rate changes is still larger than 5 s. Therefore, the overhead of feedback signaling is very small.

Fig. 7 shows the first passage times of the three transmit states. We distinguish three states according to the transmission

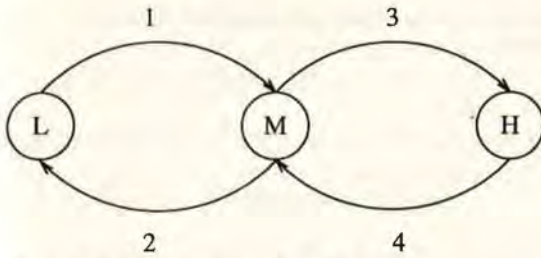


Fig. 8. FSM model for the transmission rate process.

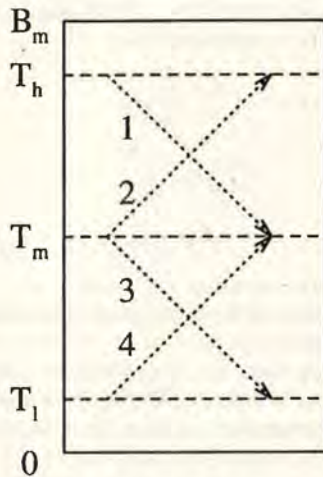


Fig. 9. Threshold crossings at the receive buffer.

rates the server chooses. The first passage time of each state denotes the average duration in which the server uses the corresponding rate. The curves show certain similarities to those in Fig. 10, that are obtained by using the analysis of the two-state Markov model, presented in the next section. However, the first passage time of medium rate state exhibits more linearity than exponentiality. Therefore, the ratio of the server using medium rate to low-high rate is almost unchanged with respect to the buffer size. The time the server spends in the medium rate state is about 4 times longer than the time spent in the low-high rate states.

These test results show the adaptiveness of the feedback mechanism in accommodating the fluctuations. The intuitive choices for the values of the thresholds and the rates are justified by the relatively smooth changes of the buffer occupancy and the transmit rates.

V. ANALYSIS OF THE RATE CONTROL PROCESS

In this section, we analyze the performance of the large receive buffer. In particular, we derive the mean duration between threshold crossings by using a first passage time analysis. The frequency of the rate change requests by the client to the server determines the amount of signaling required by the rate control protocol. The fraction of time the server is transmitting at each rate is also computed. Finally, a numerical example is presented for a two-state Markov chain model of the client decoding algorithm for VBR video service.

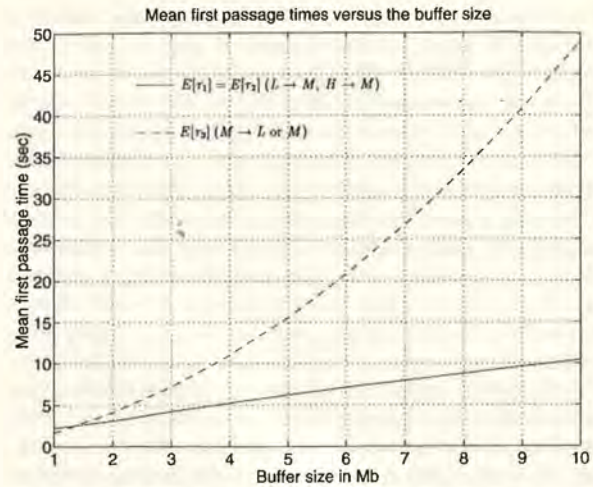


Fig. 10. Mean first passage times as a function of buffer size.

The choice of the three buffer thresholds as well as the transmission rates determine the frequency of rate change requests to maintain satisfactory performance for the multimedia service. For a fixed buffer size, if the thresholds are chosen at large distances from each other, the amount of signaling will be low at the expense of high overflow and underflow probability. The statistics of the rates used by the client decoding algorithm is another factor in selecting the thresholds. For example, for a VBR video application containing long periods of high and/or slow motion, the thresholds should be closer to prevent frequent buffer overflow and/or underflow. On the other hand, for a video sequence having bursts of high motion, closer thresholds will cause unnecessary signaling due to frequent threshold crossings.

In the following analysis, the multimedia arrival process is modeled as a fluid flow generated by a finite-state machine (FSM) having three states: Low(*L*), Medium(*M*) and High(*H*). When the FSM is in state *L*, *M*, or *H*, the arriving bit stream is a constant rate fluid flow with rate λ_l , λ_m , or λ_h , respectively, with $\lambda_l < \lambda_m < \lambda_h$. The state transition diagram of the FSM is given in Fig. 8 with each state transition labeled from one to four.

The client buffer has size B_M and three thresholds: T_l , T_m , and T_h with $T_l < T_m < T_h$. These thresholds are used by the client to produce feedback information to the server asking to change the rate of the multimedia flow to prevent buffer overflow and underflow. All possible threshold crossings are given in Fig. 9 with each transition labeled from one to four. Each threshold crossing in Fig. 9 has the same label with the corresponding state transition of the server transmission process given in Fig. 8.

The rate at which a user removes bits from the buffer is modeled by a fluid flow process modulated by a continuous time Markov chain with state transition rate matrix Q . The Markov modulated fluid flow model has been widely used to model the cell arrival process in ATM networks since its introduction in [22]. When the chain is in state *s*, information is removed at a rate R_s (in bits/s) from the buffer. The Markov

chain model captures the time correlation of rate processes generated by many multimedia sources such as VBR video encoders like MPEG. The Markov chain model of a rate process can be obtained by using the statistics of the process both in time and frequency domains. By looking the histogram of the rate process showing the frequency of each rate, one can identify clusters and assign a state to each cluster with the rate of the state equal to average rate of the cluster. The relative frequency of each cluster gives the steady state probability of the corresponding state. The time behavior of the rate process is used to obtain which transitions between clusters are possible. The average time rate stays within a cluster gives the sum of the transition rates out of the corresponding state.

In multimedia applications, a user may remove data from the buffer in batches at discrete times. As an example, an MPEG video decoder extracts each frame individually. If the thresholds are well separated, however, the time between threshold crossings will be much larger than the time between frame removals. For the computation of mean threshold crossing times, it is unnecessary to concentrate on the detailed behavior of frames. In the larger time scale, the bits are removed almost continuously. Thus, fluid flow model applies for the analysis of the events occurring in this time scale.

The following analysis can be incorporated into the client rate control protocol that adaptively changes the system parameters to achieve the desired performance during the course of a service. The client uses a Markov chain model to characterize the rate process. The parameters of the model are updated by monitoring the rate process through a time window. The buffer thresholds and transmission rates to attain a satisfactory performance are obtained by using the analysis. The transmission rates are periodically updated by sending the new values to the server.

A. First Passage Time Analysis

We analyze the expected time between threshold crossings. We assume that the feedback information is immediately available at the server. For local area networks, propagation delay is small compared to the mean period between rate changes and thus can be neglected.

We define the following three different first passage times:

- τ_1 is the first passage time from T_h to T_m (corresponding to state transition $L \rightarrow M$).
- τ_2 is the first passage time from T_l to T_m (corresponding to state transition $H \rightarrow M$).
- τ_3 is the first passage time from T_m to either T_h or T_l (corresponding to state transitions $M \rightarrow L$ or $M \rightarrow H$, respectively).

In order to find the mean values of τ_1 , τ_2 , and τ_3 , we define the following first passage time:

$$\tau = \min\{t : x(t) = B_1 \text{ or } x(t) = B_2 \text{ with } x(0) = x, S(t) = s, \text{ arrival rate } \lambda, B_1 < B_2\}$$

where $x(t)$ and $S(t)$ denote respectively the length of the buffer and the state of the Markov chain at time t . We define $E[\tau | s, x]$ as the expected value of τ starting at state (s, x) .

$E[\tau_i | s]$, $i = 1, 2, 3$ can be computed from $E[\tau | s, x]$ as follows:

- $E[\tau_1 | s] = E[\tau | s, T_h]$ with $B_1 = T_m$, $B_2 = \infty$ and $\lambda = \lambda_l$.
- $E[\tau_2 | s] = E[\tau | s, T_l]$ with $B_1 = -\infty$, $B_2 = T_m$ and $\lambda = \lambda_h$.
- $E[\tau_3 | s] = E[\tau | s, T_m]$ with $B_1 = T_l$, $B_2 = T_h$ and $\lambda = \lambda_m$.

$E[\tau | s, x] = 0$ for $x \geq B_2$ and $x \leq B_1$. For $B_1 < x < B_2$ and starting at a state s with $\lambda > R_s$, the queue length will hit the upper threshold B_2 if the Markov chain stays at s for longer than $(B_2 - x)/(\lambda - R_s)$. If the state changes earlier, $E[\tau | s, x]$ can be written recursively

$$E[\tau | s, x] = e^{-q_s \frac{B_2 - x}{\lambda - R_s}} \frac{B_2 - x}{\lambda - R_s} + \sum_{s' \neq s} q_{ss'} \int_0^{\frac{B_2 - x}{\lambda - R_s}} e^{-q_s t} \cdot [t + E[\tau | s', x + (\lambda - R_s)t]] dt \quad (1)$$

where $q_{ss'}$ is the transition rate from s to s' and $q_s = \sum_{s' \neq s} q_{ss'}$ is the rate of the exponential distribution governing the time spent at s .

When the chain starts at a state s with $\lambda < R_s$, the buffer length will hit the lower threshold B_1 if no transition occurs from s for a duration of at least $(x - B_1)/(R_s - \lambda)$ and $E[\tau | s, x]$ can be written recursively as

$$E[\tau | s, x] = e^{-q_s \frac{x - B_1}{R_s - \lambda}} \frac{x - B_1}{R_s - \lambda} + \sum_{s' \neq s} q_{ss'} \int_0^{\frac{x - B_1}{R_s - \lambda}} e^{-q_s t} \cdot [t + E[\tau | s', x + (\lambda - R_s)t]] dt. \quad (2)$$

We define the Laplace transform $F_s(\theta)$ of $E[\tau | s, x]$ with respect to the initial buffer length as

$$F_s(\theta) = \int_0^\infty E[\tau | s, x] e^{-\theta x} dx \quad (3)$$

and define the N -dimensional vector

$$\underline{F}(\theta) = [F_1(\theta), \dots, F_N(\theta)]^T \quad (4)$$

where N is the number of states in the Markov chain. We prove in Appendix B that $\underline{F}(\theta)$ can be written as

$$\underline{F}(\theta) = (Q + \theta D)^{-1} \underline{b}(\theta) \quad (5)$$

where $D = \text{diag}\{\lambda - R_s\}$ is the drift matrix and $\underline{b}(\theta)$ as defined in Appendix B is an N -dimensional vector. With the rate control process, the probability of buffer overflow and underflow will be very small and hence the boundary conditions at zero and B_M are neglected in the analysis.

The inverse Laplace transform can be performed by using partial fraction expansion. As shown in Appendix B, we can write

$$\underline{b}(\theta) = e^{-\theta B_1} \underline{b}_1 + e^{-\theta B_2} \underline{b}_2$$

where b_1 and b_2 are obtained by applying boundary conditions at B_1 and B_2 . Let $\{\theta_k\}_{k=1}^N$ be the roots of the equation

$$\det(Q + \theta D) = 0. \quad (6)$$

Since all row sums of Q are zero, $\theta = 0$ is a root of (6), say $\theta_N = 0$. We observe that both b_1 and b_2 contain the factor $1/\theta$. If all the roots of (6) are distinct, we write

$$\begin{aligned} \underline{F}(\theta) = & \left[\sum_{k=1}^{N-1} \frac{A_k}{\theta - \theta_k} + \frac{A_N}{\theta^2} + \frac{A_{N+1}}{\theta} \right] e^{-\theta B_1} \\ & + \left[\sum_{k=1}^{N-1} \frac{B_k}{\theta - \theta_k} + \frac{B_N}{\theta^2} + \frac{B_{N+1}}{\theta} \right] e^{-\theta B_2} \end{aligned} \quad (7)$$

where

$$\begin{aligned} A_k &= \lim_{\theta \rightarrow \theta_k} (\theta - \theta_k)(Q + \theta D)^{-1} b_1 \\ A_N &= \lim_{\theta \rightarrow 0} \theta^2 (Q + \theta D)^{-1} b_1 \\ A_{N+1} &= \lim_{\theta \rightarrow 0} \frac{\partial}{\partial \theta} (\theta^2 (Q + \theta D)^{-1}) b_1 \end{aligned}$$

and $\{B_k\}_{k=1}^{N+1}$ are similarly defined with b_1 replaced by b_2 . If there are repeated roots of (6), the partial fraction expansion can be obtained accordingly.

Since we are interested in the inverse Laplace transform within the region $B_1 \leq x \leq B_2$, we perform the operation for only the first term in (7). The application of the boundary conditions in obtaining b_1 and b_2 nullifies the solution outside $B_1 \leq x \leq B_2$. Let $\underline{H}(x) = [E[\tau | 1, x], \dots, E[\tau | N, x]]^T$ be the inverse Laplace transform of $\underline{F}(\theta)$. We have for $B_1 \leq x \leq B_2$

$$\begin{aligned} \underline{H}(x) = & \sum_{k=1}^{N-1} A_k e^{\theta_k(x-B_1)} + A_N \frac{(x-B_1)^2}{2} \\ & + A_{N+1}(x-B_1). \end{aligned} \quad (8)$$

The complexity of the main computations in obtaining $\underline{H}(x)$ is in the order of N^3 by the standard Gaussian elimination for the solutions of b_1 and b_2 .

After calculating the mean threshold crossing times, the relative frequency of each rate can also be obtained. Let f_l , f_h , and f_m be the proportion of time each rate is used. Obviously

$$f_l + f_h + f_m = 1. \quad (9)$$

Furthermore, the average rate, R_{av} is given by

$$R_{av} = f_l \lambda_l + f_h \lambda_h + f_m \lambda_m. \quad (10)$$

Consider a long period t . Let n_h and n_l respectively be the number of times T_l and T_h are crossed during t . Since each low or high rate period is followed by a period with medium rate $n_l(E[\tau_1] + E[\tau_3]) + n_h(E[\tau_2] + E[\tau_3]) = t$. The average rate over period t is R_{av} , i.e., $n_l(E[\tau_1]\lambda_l + E[\tau_3]\lambda_m) + n_h(E[\tau_2]\lambda_h + E[\tau_3]\lambda_m) = R_{av}t$. Combining the last two equations we have

$$\frac{n_l}{n_h} = \frac{E[\tau_2](\lambda_h - R_{av}) + E[\tau_3](\lambda_m - R_{av})}{E[\tau_1](R_{av} - \lambda_l) + E[\tau_3](R_{av} - \lambda_m)}. \quad (11)$$

On the other hand

$$\frac{f_l}{f_h} = \frac{n_l E[\tau_1]}{n_h E[\tau_2]}. \quad (12)$$

We obtain f_l , f_h , and f_m by solving (9)–(12).

The analysis technique developed in this section is applicable to any Markov chain model. In the following subsection, as an example, we derive the solutions for the mean first passage times in detail for a two-state Markov chain.

B. Two-State Markov Chain Model

We model the process describing the rate at which the bits are removed from the receive buffer with a two-state Markov chain. For video encoding schemes that use only interframe technique without motion estimation, a two-state model can be applicable. A video sequence containing periods of high and low motion can be modeled by two states, one state with low bitrate (corresponding to low motion) and the other state with high bitrate (corresponding to high motion). Let

$$Q = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix}$$

be the state transition rate matrix of the Markov chain. The parameters $1/\alpha$ and $1/\beta$ are matched to average durations of high and low motion periods. Without loss of generality assume that $R_1 < R_2$. The roots $\theta_{1,2}$ of (6) are given by

$$\theta_1(\lambda) = \frac{\beta(\lambda - R_1) + \alpha(\lambda - R_2)}{(\lambda - R_1)(\lambda - R_2)}, \quad \theta_2(\lambda) = 0.$$

The high rate λ_h is chosen to be larger than both R_1 and R_2 in order to have a drift toward T_m after hitting T_l , whereas the low rate λ_l is similarly selected smaller than both R_1 and R_2 . The medium rate λ_m is chosen as the average rate, $\lambda_m = (R_1\beta + R_2\alpha)/(\alpha + \beta)$. Hence, we have

$$\lambda_l < R_1 < \lambda_m < R_2 < \lambda_h.$$

We find the mean values of τ_1 , τ_2 , and τ_3 separately.

1) $E[\tau_1]$ (mean first passage time from L to M): Since $R_1 < \lambda_m < R_2$, the chain is in state 1 when T_h is hit, i.e., $E[\tau_1] = E[\tau_1 | 1]$. By doing the partial fraction expansion in (7) and taking the inverse Laplace transform with respect to θ , with $\text{Re}\{\theta\} > 0$, we obtain

$$\begin{aligned} E[\tau_1] = & -\frac{(\alpha + \beta)(T_h - T_m)}{\beta(\lambda_l - R_1) + \alpha(\lambda_l - R_2)} \\ & + \frac{\alpha(R_1 - R_2)(\lambda_l - R_2)}{[\beta(\lambda_l - R_1) + \alpha(\lambda_l - R_2)]^2} (1 - e^{\theta_1(\lambda_l)(T_h - T_m)}). \end{aligned} \quad (13)$$

The first term in (13) corresponds to the steady-state behavior of the Markov chain, whereas the second term reflects the effect of the initial state.

2) $E[\tau_2]$ (mean first passage time from H to M): Since $R_1 < \lambda_m < R_2$, the chain is in state 2 when T_l is hit, i.e., $E[\tau_2] = E[\tau_2 | 2]$. By doing the partial fraction expansion in (7) and taking the inverse Laplace transform with respect to θ , with $\text{Re}\{\theta\} < 0$, we obtain

$$E[\tau_2] = \frac{(\alpha + \beta)(T_m - T_l)}{\beta(\lambda_h - R_1) + \alpha(\lambda_h - R_2)} + \frac{\beta(R_2 - R_1)(\lambda_h - R_1)}{[\beta(\lambda_h - R_1) + \alpha(\lambda_h - R_2)]^2} (1 - e^{\theta_1(\lambda_h)(T_l - T_m)}) \quad (14)$$

3) $E[\tau_3]$ (mean first passage time from M to either L or H): The state of the chain at the beginning can be either one or two. Since $R_1 < \lambda_m < R_2$, we need to evaluate the boundary conditions. Solving the linear system of equations we obtain the two unknown constants

$$\gamma_1 = \frac{(\alpha + \beta)(T_h - T_l)^2/2 - (\lambda_m - R_2)(T_h - T_l)}{\beta(T_h - T_l) - (\lambda_m - R_2)}$$

$$\gamma_2 = \frac{(\alpha + \beta)(T_h - T_l)^2/2 - (\lambda_m - R_1)(T_h - T_l)}{\lambda_m - R_1 - \alpha(T_h - T_l)}$$

Performing the partial fraction expansion and taking the inverse Laplace transform we obtain (15) and (16), see equations at the bottom of this page.

Example: We choose the following parameters for the two-state Markov chain

$$\alpha = \beta \cong 0.2 \text{ s}^{-1}$$

and

$$R_1 = 0.7 \text{ Mb/s}, R_2 = 1.3 \text{ Mb/s}.$$

These parameters correspond to a video sequence containing low and high motion of equal amount and each low or high period last for 5 s on the average. The average rate is 1.0 Mb/s. We choose the following transmission rates

$$\lambda_l = 0.5 \text{ Mb/s}, \lambda_m = 1.0 \text{ Mb/s}, \lambda_h = 1.5 \text{ Mb/s}.$$

The thresholds T_l , T_m , and T_h are chosen as

$$T_l = 0.1B_M, T_m = 0.5B_M \text{ and } T_h = 0.9B_M$$

and B_M varies from 1 Mb/s to 10 Mb/s (1 s to 10 s of video). The mean first passage times are shown in Fig. 10 as a function

of buffer size, where the solid curve represents $E[\tau_1] = E[\tau_2]$ and the dashed curve corresponds to $E[\tau_3]$. Although the parameters used in the plots in Fig. 10 are not matched to the specific rate process used for the experimentation in Section IV, the curves show similar trends with the curves in Fig. 7.

When B_M is large, $E[\tau_1]$ and $E[\tau_2]$ are dominated by the first term in (13) and (14) respectively. The reason for this is that as B_M gets larger, the mean first passage times also increase and the effect of the initial state is washed out.

When the transmission rate is λ_m , the average drift is zero. $E[\tau_3]$ is determined by the occurrence of a temporary overload or underload period whose duration should increase as B_M gets larger. Since the time spent at each state is exponentially distributed, the probability of a long overload or underload period decreases exponentially as B_M increases. Therefore, $E[\tau_3]$ increases exponentially with the buffer size. For large buffer size, starting at the medium threshold, the mean time to hit the lower or upper threshold is large compared to mean passage times from lower and upper thresholds to the medium threshold. Therefore, f_m is much larger than f_l and f_h when the receive buffer is large (at $B_M = 10 \text{ Mb}$, $f_m = 5/6$ and $f_l = f_h = 1/12$).

VI. CONCLUSION

We have demonstrated the adaptiveness and efficiency of client-based rate and synchronization control through the GRAMS implementation. Currently, we are refining the interface and performing further traffic studies based on real applications. Also, we presented new performance analysis methodologies using first passage time analysis, for which, both the state of the buffer and the state of the video encoding algorithm are considered.

The current GRAMS server as well as the client interface and control are functioning as expected in our current implementation. Further experience will be gained and reported as the server and client are deployed locally on our campus.

APPENDIX A IMPLEMENTATION OF MULTIMEDIA SERVER AND CLIENT INTERFACE

In GRAMS, three types of media are distinguished and a queue is associated with each media type. In decreasing order of service priority, they are as follows:

Continuous real-time media: This includes video (CBR and VBR video) and audio that requires continuous transmission and playback. The major QoS concerns are latency and jitter as defined earlier.

$$E[\tau_3 | 1] = \frac{(\lambda_m - R_2)\gamma_1 - (\beta\gamma_1 + \lambda_m - R_2)(T_m - T_l) + (\alpha + \beta)(T_m - T_l)^2/2}{(\lambda_m - R_1)(\lambda_m - R_2)} \quad (15)$$

$$E[\tau_3 | 2] = \frac{(\lambda_m - R_1)\gamma_2 + (\lambda_m - R_1 - \alpha\gamma_2)(T_m - T_l) - (\alpha + \beta)(T_m - T_l)^2/2}{(\lambda_m - R_1)(\lambda_m - R_2)} \quad (16)$$

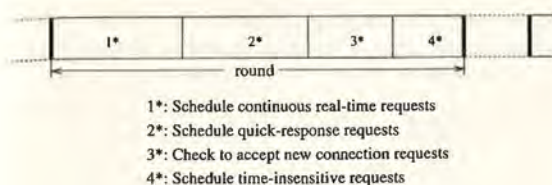


Fig. 11. Basic frame structure.

Quick response data: This includes images, postscript files for transparencies, etc., for which the major QoS concern is latency.

Time insensitive data: This includes all sorts of retrieval data that do not have strict delay requirements.

The service of these three queues is done in a round robin fashion in rounds of fixed duration (Fig. 11). A packet is sent for each connection in a round. The rate of the connection is therefore given by the packet size divided by the duration of the round. Rate assignment is done by packet size assignment during connection setup. Rate adjustment by changing the packet size occurs when feedback requests are made downstream by the client. The packets are served according to the priority of the media type described above. Transmission starvation (or jitter) occurs for a connection when a packet for that connection is not transmitted within that round. Cache starvation occurs when a packet cannot be transmitted in a round due to cache underflow.

We monitor the QoS at the server by computing the underflow jitter rate of the transmit value. The starvation count is updated per round for each connection i according to

$$C_i \leftarrow \alpha_i C_i + I_i \quad (17)$$

in which $I_i = 1$ if, during the round no packet is sent due to cache underflow or transmission starvation. Compensation of transmission may occur if the round has time left over and if C_i is positive. In this case, I_i is equal to one minus the number of packets transmitted for the connection i . For quick response data services such as image transport, we set $\alpha_i = 1$ so that C_i represents the cumulative transmission deficit. For video transmission, we set $\alpha_i < 1$ so that C_i represents a geometrically weighted running sum of the transmission deficit.

The single parameter C_i keeps track of the QoS, for which, the server is trying to maintain for connection i . However, the actual QoS perceived by the client is better than C_i because further buffering and smoothing of jitter is performed at the client. Nevertheless, the local server QoS parameter is useful because the degradation of this QoS parameter can result from usage congestion of any resource, such as; the disk bus, cache memory, transmission bandwidth, or processing cycles. When the parameter exceeds a certain threshold, GRAMS attempts to reduce usage congestion by exercising rate control and admission control. Rate control reduces the rates of quick response data by actions such as reducing the packet size for images. Admission control delays new admissions until the QoS parameter recedes. Fig. 12 plots the overall system starvation rate as a function of the offered load, α , and thresholds for video. As expected, starvation

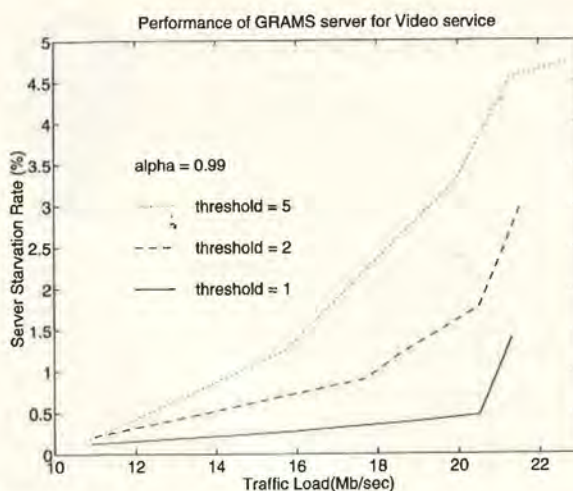


Fig. 12. Server starvation rate.

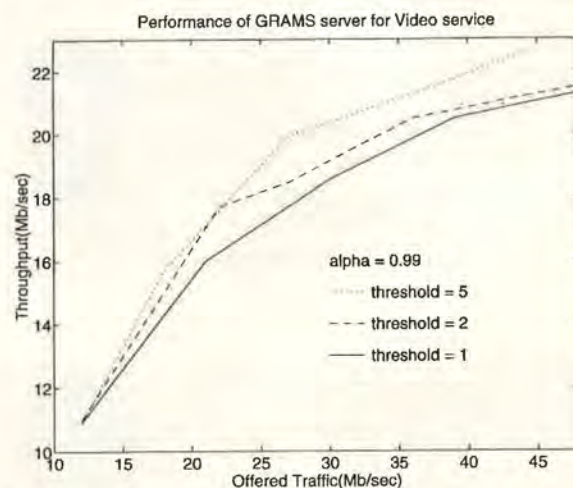


Fig. 13. Server throughput.

increases as offered load is increased. Starvation can occur even at low load because of the rather unpredictable load in each round for the operating system, making available processing cycles in each round for caching and transmission functions somewhat unpredictable. Such starvation has the effect of gating admission very much the same way as a traffic light. Fig. 13 plots the carried load as a function of the offered load (assuming failed admission are cleared instead of being queued). The difference between carried load and the offered load reflects the fraction of connection requests that suffers delay (often minimal) before the connection is completed. Thus, we have a system that is adaptive and degrades gracefully as system load is increased, and requires very little knowledge of traffic statistics.

APPENDIX B

In this Appendix, we derive the Laplace transform $F_s(\theta)$ of $E[\tau | s, x]$ with respect to x and show that the N -dimensional vector $\underline{F}(\theta)$ given by (4) can be represented by (5).

Case 1: $\lambda > R_s$

Taking the Laplace transform of (1), we have

$$\begin{aligned} F_s(\theta) &= \int_{B_1}^{B_2} e^{-\theta x} e^{-q_s \frac{B_2-x}{\lambda-R_s}} \frac{B_2-x}{\lambda-R_s} dx + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x} \\ &\quad \cdot \int_0^{\frac{B_2-x}{\lambda-R_s}} e^{-q_s t} [t + E[\tau | s', x + (\lambda - R_s)t]] dt dx \\ &= \frac{1}{q_s} \int_{B_1}^{B_2} e^{-\theta x} \left(1 - e^{-q_s \frac{B_2-x}{\lambda-R_s}}\right) dx \\ &\quad + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x'} E[\tau | s', x'] \\ &\quad \cdot \int_0^{\frac{x'-B_1}{\lambda-R_s}} e^{-(q_s - \theta(\lambda-R_s))t} dt dx' \end{aligned}$$

where we used the change of variable $x' = x + (\lambda - R_s)t$ to obtain the second term.

$$\begin{aligned} F_s(\theta) &= \frac{1}{q_s} \left[\frac{1}{\theta} (e^{-\theta B_1} - e^{-\theta B_2}) + \frac{(\lambda - R_s) e^{-\frac{q_s B_2}{\lambda-R_s}}}{q_s - \theta(\lambda - R_s)} \right. \\ &\quad \cdot \left(e^{-(\theta - \frac{q_s}{\lambda-R_s}) B_1} - e^{-(\theta - \frac{q_s}{\lambda-R_s}) B_2} \right) \Big] \\ &\quad + \frac{1}{q_s - \theta(\lambda - R_s)} \sum_{s' \neq s} q_{ss'} \\ &\quad \cdot [F_{s'}(\theta) - e^{-(\theta - \frac{q_s}{\lambda-R_s}) B_1} \\ &\quad \cdot \int_{B_1}^{B_2} E[\tau | s', x'] e^{-\frac{q_s}{\lambda-R_s} x'} dx'] \\ &= \frac{1}{\theta q_s (q_s - \theta(\lambda - R_s))} [e^{-\theta B_1} (q_s - \theta(\lambda - R_s)) \\ &\quad \cdot (1 - e^{-\frac{q_s}{\lambda-R_s} (B_2 - B_1)}) - e^{-\theta B_2} q_s] \\ &\quad + \frac{1}{q_s - \theta(\lambda - R_s)} \sum_{s' \neq s} q_{ss'} \\ &\quad \cdot \left[F_{s'}(\theta) - e^{-(\theta - \frac{q_s}{\lambda-R_s}) B_1} F_{s'} \left(\frac{q_s}{\lambda - R_s} \right) \right]. \quad (18) \end{aligned}$$

Case 2: $\lambda < R_s$

Taking the Laplace transform of (2), we have

$$\begin{aligned} F_s(\theta) &= \int_{B_1}^{B_2} e^{-\theta x} e^{-q_s \frac{x-B_1}{R_s-\lambda}} \frac{x-B_1}{R_s-\lambda} dx \\ &\quad + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x} \int_0^{\frac{x-B_1}{R_s-\lambda}} e^{-q_s t} \\ &\quad \cdot [t + E[\tau | s', x + (\lambda - R_s)t]] dt dx \\ &= \frac{1}{q_s} \int_{B_1}^{B_2} e^{-\theta x} \left(1 - e^{-q_s \frac{x-B_1}{R_s-\lambda}}\right) dx \\ &\quad + \sum_{s' \neq s} q_{ss'} \int_{B_1}^{B_2} e^{-\theta x'} E[\tau | s', x'] \\ &\quad \cdot \int_0^{\frac{B_2-x'}{R_s-\lambda}} e^{-(q_s - \theta(\lambda-R_s))t} dt dx' \end{aligned}$$

where we used the change of variable $x' = x + (\lambda - R_s)t$ to obtain the second term. Evaluating the above integrals we have

$$\begin{aligned} F_s(\theta) &= \frac{1}{\theta q_s (q_s - \theta(\lambda - R_s))} \left[e^{-\theta B_1} q_s - e^{-\theta B_2} \right. \\ &\quad \cdot \left(q_s - \theta(\lambda - R_s) \left(1 - e^{-\frac{q_s}{R_s-\lambda} (B_2 - B_1)}\right) \right) \Big] \\ &\quad + \frac{1}{q_s - \theta(\lambda - R_s)} \sum_{s' \neq s} q_{ss'} \\ &\quad \cdot \left[F_{s'}(\theta) - e^{-(\theta - \frac{q_s}{\lambda-R_s}) B_2} F_{s'} \left(\frac{q_s}{\lambda - R_s} \right) \right]. \quad (19) \end{aligned}$$

Comparing (18) and (19) we observe that both equations have the same form with B_1 and B_2 interchanged. Multiplying both sides of (18) and (19) with $q_s - \theta(\lambda - R_s)$ and arranging the terms, we obtain the matrix equation

$$(Q + \theta D) \underline{F}(\theta) = \underline{b}(\theta)$$

where $\underline{b}(\theta) = [b_1(\theta), \dots, b_N(\theta)]^T$ is given by

$$\begin{aligned} b_s(\theta) &= \frac{1}{\theta q_s} \left[-e^{-\theta B_1} (q_s - \theta(\lambda - R_s)) \right. \\ &\quad \cdot \left(1 - e^{-\frac{q_s}{\lambda-R_s} (B_2 - B_1)} \right) + e^{-\theta B_2} q_s \Big] \\ &\quad + e^{-(\theta - \frac{q_s}{\lambda-R_s}) B_1} \sum_{s' \neq s} q_{ss'} F_{s'} \left(\frac{q_s}{\lambda - R_s} \right) \quad (20) \end{aligned}$$

for a state s with $\lambda > R_s$. For s with $\lambda < R_s$, $b_s(\theta)$ is the same as (20) with B_1 and B_2 interchanged. $b_s(\theta)$ contains $N(N-1)$ unknown constants $\{F_{s'}(\frac{q_s}{\lambda-R_s})\}_{s \neq s'}$.

These constants correspond to the discontinuity of $E[\tau | s, x]$ at B_1 or B_2 . For a state s with $\lambda > R_s$

$$\lim_{x \nearrow B_2} E[\tau | s, x] = 0$$

whereas

$$\lim_{x \searrow B_1} E[\tau | s, x] > 0$$

although $E[\tau | s, B_1] = 0$. Similarly, for a state s with $\lambda < R_s$, the discontinuity occurs at B_2 . We can write for s with $\lambda > R_s$

$$b_s(\theta) = -\frac{1}{\theta} (e^{-\theta B_1} - e^{-\theta B_2}) + \gamma_s e^{-\theta B_1} \quad (21)$$

and for a state s with $\lambda < R_s$

$$b_s(\theta) = -\frac{1}{\theta} (e^{-\theta B_1} - e^{-\theta B_2}) + \gamma_s e^{-\theta B_2} \quad (22)$$

where the unknown constants $\{\gamma_s\}$ can be expressed from (20). These constants are evaluated by taking the inverse Laplace transform of $\underline{F}(\theta) = (Q + \theta D)^{-1} \underline{b}(\theta)$ and applying the following boundary conditions: For each state s with $\lambda > R_s$, we set $E[\tau | s, B_2] = 0$ and for s with $\lambda < R_s$, we set $E[\tau | s, B_1] = 0$. Since $\{\gamma_s\}$ appears as a coefficient in the corresponding mean passage time equations, $\{\gamma_s\}$ can be obtained as the solution of a system of N linear equations.

REFERENCES

- [1] T. D. C. Little and F. Kao, "An intermedia skew control system for multimedia data presentation," in *Proc. 3rd Int. Workshop Network Operating Syst. Support Digital Audio Video*, La Jolla, CA, Nov. 1992.
- [2] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, Apr. 1990.
- [3] ———, "Multimedia synchronization protocols for broadband integrated services," *IEEE J. Select. Areas Commun.*, vol. 9, no. 9, Dec. 1991.
- [4] N. U. Qazi, M. Woo, and A. Ghafoor, "A synchronization and communication model for distributed multimedia objects," in *Proc. ACM Multimedia*, Anaheim, CA, 1993.
- [5] P. V. Rangan, S. Ramanathan, and H. M. Vin, "Techniques for multimedia synchronization in network file systems," *IEEE Comp. Commun.*, vol. 16, no. 3, Mar. 1993.
- [6] P. V. Rangan and H. M. Vin, "Designing an on-demand multimedia service," *IEEE Commun. Mag.*, vol. 30, no. 7, July 1992.
- [7] H. P. Katseff and B. S. Robinson, "Predictive pre-fetch in the nemesi multimedia information service," in *Proc. ACM Multimedia*, San Francisco, Oct. 1994.
- [8] P. Hoepner, "Synchronizing the presentation of multimedia objects," *IEEE Commun. Mag.*, vol. 15, no. 9, Nov. 1992.
- [9] R. Steinmetz, "Synchronization properties in multimedia systems," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, Apr. 1990.
- [10] D. P. Anderson and G. Homsy, "A continuous media I/O server and its synchronization mechanism," *IEEE Comp.*, vol. 24, no. 10, Oct. 1991.
- [11] D. Shepherd, D. Hutchinson, F. Garcia, and G. Coulson, "Protocol support for distributed multimedia applications," *Comp. Commun.*, vol. 13, no. 7, Sept. 1990.
- [12] J. Y. Hui, J. Zhang, and J. Li, "Quality of service control in GRAMS for ATM local area network," *IEEE J. Select. Areas Commun.*, vol. 13, no. 4, May 1995.
- [13] D. Minoli and R. Keinath, *Distributed Multimedia Through Broadband Communications Services*. Norwood, MA: Artech House, 1994.
- [14] J. Y. Hui, "Resource allocation for broadband networks," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, Dec. 1988.
- [15] A. Pitsillides and J. Lambert, "Adaptive connection admission and flow control: Quality of service with high utilization," in *Proc. Conf. Comp. Commun. (IEEE Infocom)*, Toronto, Canada, June 1994.
- [16] G. Ramamurthy and R. S. Dighe, "Distributed source control: a network access control for integrated broadband packet networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 990–1002, Sept. 1991.
- [17] K. Obraczka et al., "Internet resource discovery services," *IEEE Comput.*, pp. 8–22, Sept. 1993.
- [18] G. Branwyn, *Mosaic Quick Tour: Accessing & Navigating the Internet's World Wide Web*. Chapel Hill, NC: Ventana Press, 1994.
- [19] SunSoft Incorporated, "CellB Codec," in *Imaging Library Programmer's Guide (Solaris XIL 1.1 Imaging Library AnswerBook)*.
- [20] W. Stallings, *Advances in ISDN and Broadband ISDN*. Los Alamitos, CA: IEEE Computer Society Press, 1992.
- [21] D. Reininger, B. Melamed, and D. Raychaudhuri, "Variable bitrate MPEG video: characteristics, modeling and multiplexing," in *Proc. 14th Int. Teletraffic Congress*, Antibes Juan-les-Pins, France, June 6–10, 1994.
- [22] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of data handling systems with multiple sources," *Bell Syst. Tech. J.*, vol. 61, no. 8, Oct. 1982.



Joseph Y. Hui (S'82–M'83–SM'90) received the S.B., S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1981, 1981, 1982, and 1983, respectively.

He joined Bell Communications Research, Morristown, NJ, in 1983, and Rutgers University, New Brunswick, NJ, in 1989 as an Associate Professor in the Department of Electrical and Computer Engineering. From 1979 to 1981, he worked at the Comsat Laboratories under the MIT Cooperative Program. He was also an Adjunct Professor at the Center for Telecommunications Research, Columbia University, New York, NY, from 1987 to 1988. His current research interests include coding and information theory, wireless information networks, and switching, protocol, traffic, and security issues for integrated broadband networks. He is the author of a book entitled *Switching and Traffic Theory for Integrated Broadband Networks* (Kluwer Academic, 1990).

Dr. Hui is the recipient of the 1985 William Bennett Prize Paper Award given by the IEEE Communications Society. His work on multiple access communications and switching networks resulted in several patents. He also received the Presidential Young Investigator Award from the National Science Foundation.



Ezhan Karasan (S'89) received the B.S. degree in electrical engineering from the Middle East Technical University, Ankara, Turkey and the M.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 1987 and 1990, respectively. He is currently pursuing his Ph.D. degree in electrical engineering at Rutgers University, New Brunswick, New Jersey.

His research interests include performance analysis of broadband networks and multimedia systems.

Mr. Karasan was awarded a fellowship from the NATO Science Scholarship Program from 1991 to 1994.



Jun Li (S'93) received the B.Sc. and M.Sc. degrees in electrical engineering from Xidian University, Xi'an, PRC, in 1983 and 1985, respectively. He is currently a Ph.D. student in the ECE department of Rutgers University, specializing in wireless multimedia services and broadband networks.

He worked as an assistant professor at Beijing Information Technology Institute, Beijing, PRC, from 1986 to 1990, working on speech synthesis and recognition. He served as a visiting researcher at NEC's C&C Central Labs, Kawasaki, Japan, from

1987 to 1989. From 1990 to 1993, he worked for Data Control Limited, Tokyo, Japan and San Francisco, CA, as a software engineer, developing LAN softwares for multi-platform computers.



Junbiao Zhang received the B.S. degree in computer science from the University of Science and Technology of China, in 1990, and the M.S. degree in computer science from Rutgers University, Piscataway, NJ, in 1994. He is now a Ph.D. student in the computer science department at Rutgers University. His research interests include distributed systems, high speed networks, and multimedia systems. He also designed and implemented the multimedia service system GRAMS.

Ex. 1019B

LIBRARY | CATALOG

1 of 1

PERIODICAL OR NEWSPAPER

IEEE journal on selected areas in communications : a publication of the IEEE ...

[Full Record](#) [MARC Tags](#)

Main title

IEEE journal on selected areas in communications : a publication of the IEEE Communications Society.

Published/Created

New York, NY.: Institute of Electrical and Electronics Engineers, ©1983-

[Request this item](#)

[LC Find It](#)

More Information

LCCN Permalink	https://lccn.loc.gov/833643067
Publication history	Vol. SAC-1, no. 1 (Jan. 1983)-
Description	volumes : illustrations ; 28 cm
Current frequency	Eleven times a year, <Dec. 2012->
Former frequency	Frequency varies, 1983- Ten no. a year, <2004->
ISSN	0733-8716

Linking ISSN	0733-8716
LC classification	TK5101.A1 I35
Variant title	Institute of Electrical and Electronics Engineers journal on selected areas in communications
Portion of title	Selected areas in communications
Serial key title	IEEE journal on selected areas in communications (Print)
Abbreviated title	IEEE j. sel. areas commun. (Print)
Related names	Institute of Electrical and Electronics Engineers. IEEE Communications Society.
LC Subjects	Telecommunication--Periodicals.
Other Subjects	Telecommunication.
Form/Genre	Periodicals. Periodicals.
Browse by shelf order	TK5101.A1
Notes	Title from cover.
Reproduction no./Source	IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854
Additional formats	Online version: IEEE journal on selected areas in communications (Online) 1558-0008 (DLC) 2005215206 (OCoLC)44607980 Online version: IEEE journal on selected areas in communications (OCoLC)752408668
LCCN	83643067
Invalid LCCN	sn 82005410
CODEN	ISACEM
Dewey class no.	621.38/05
Other system no.	(OCoLC)ocm08649040

Additional Links	http://ieeexplore.ieee.org/servlet/opac?punumber=49
Type of material	Periodical or Newspaper
Content type	text
Media type	unmediated
Carrier type	volume
Source of description	Latest issue consulted: Vol. 30, no. 11 (Dec. 2012).

Item Availability



CALL NUMBER	TK5101.A1 I35 Set 1
Request in	Jefferson or Adams Building Reading Rooms
Status	c.1 v. 41, no. 5 2023 May At Bindery - 01-17-2025
Older receipts	v.1 (1983), v.4:no.1-v.9:no.1 (1986:Jan.-1991:Jan.), v.9:no.4-v.14:no.6 (1991:May-1996:June), v.14:no.9-v.18:no.1 (1996:Sept.-2000:Jan.), v.18:no.3-v.19:no.9 (2000:Mar.-2001:Sept.) v.19:no.11-v.22:no.5 (2001:Nov.-2004:June), v.23:no.1-v.23:no.3 (2005:Jan.-2005:Mar.), v.23:no.5-v.25:no.6 (2005:May-2007:Aug.), v.25:no.8-v.25:no.9 (2007:Oct.-2007:Dec.), v.31:no.8-v.41:no.3 (2013:July-2023:Mar.) v.41:no.5-v.41:no.12 (2023:May-2023:Dec.)
Supplements	v.24:no.8 (2006:Aug.), v.24:no.12 (2006:Dec.), v.25:no.4 (2007:Apr.), v.25:no.6 (2007:Aug.), v.25:no.9 (2007:Dec.), v.26:no.6-v.26:no.1-v.26:no.2 (2008:Jan.-2008:Feb.), v.26:no.9-v.31:no.6 (2008:Aug.-2013:June), v.31:no.9 (2013:Sept.)

CALL NUMBER	TK5101.A1 I35 Copy 1 Unbound issues
Request in	Newspaper & Current Periodical Reading Room (Madison LM133)

Latest receipts

v. 42, no. 12 (2024 Dec.)
v. 42, no. 11 (2024 Nov.)
v. 42, no. 10 (2024 Oct.)
v. 42, no. 9 (2024 Sept.)
v. 42, no. 8 (2024 Aug.)
v. 42, no. 7 (2024 July)
v. 42, no. 6 (2024 June)
v. 42, no. 5 (2024 May)
v. 42, no. 4 (2024 Apr.)
v. 42, no. 3 (2024 Mar.)
v. 42, no. 2 (2024 Feb.)
v. 42, no. 1 (2024 Jan.)
v. 26, no. 5 (2008 June)

LIBRARY | CATALOG



1 of 1

PERIODICAL OR NEWSPAPER

IEEE journal on selected areas in communications : a publication of the IEEE ...

Full Record MARC Tags

```

000      03909cas a2200721 a 4500
001      11272751
005      20181122162625.0
008      820802c19839999nyumxp 0 a0eng c
010      — |a 83643067 |z sn 82005410
022      0_ |a 0733-8716 || 0733-8716 |2 1
030      — |a SACEM
032      — |a 690550 |b JSPS
035      — |a (OCLC)ocm08649040
037      — |b IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854
040      — |a VSD |b eng |c VSD |d DLC |d NST |d DLC |d NST |d AIP |d NST |d AIP |d NST |
      |d VSD |d NST |d OCL |d MYG |d OCL |d CUS |d OCL |d MYG |d IEEE X |d OCLCQ |
      |d DLC |d OCLCQ |d NSD |d LVB |d CIT |d OCLCQ |d DLC |d OCLCF |d OCLCO |d IUL |
      |d OCLCA |d GILDS |d OCLCO |d OCLCQ
042      — |a ooc |a nsdp
050      00 |a TK5101 .A1 |b I35
082      04 |a 621.3805 |2 19
210      0_ |a IEEE j. sel. areas commun. |b (Print)
222      _0 |a IEEE journal on selected areas in communications |b (Print)
245      00 |a IEEE journal on selected areas in communications : |b a publication of the IEEE
      Communications Society.
246      3_ |a Institute of Electrical and Electronics Engineers journal on selected areas in

```

communications

- 246 30 |a Selected areas in communications
- 260 — |a New York, NY. : |b Institute of Electrical and Electronics Engineers, |c ©1983-
- 300 — |a volumes : |b illustrations ; |c 28 cm
- 310 — |a Eleven times a year, |b <Dec. 2012->
- 321 — |a Frequency varies, |b 1983-
- 321 — |a Ten no. a year, |b <2004->
- 336 — |a text |b txt |2 rdacontent
- 337 — |a unmediated |b n |2 rdamedia
- 338 — |a volume |b nc |2 rdacarrier
- 362 0_ |a Vol. SAC-1, no. 1 (Jan. 1983)-
- 500 — |a Title from cover.
- 588 — |a Latest issue consulted: Vol. 30, no. 11 (Dec. 2012).
- 590 — |a SERBIB/SERLOC merged record
- 650 _0 |a Telecommunication |v Periodicals.
- 650 _6 |a Télécommunications |v Périodiques.
- 650 _7 |a Telecommunication. |2 fast |0 (OCoLC)fst01145830
- 655 _7 |a Periodicals. |2 fast |0 (OCoLC)fst01411641
- 655 _7 |a Periodicals. |2 lcgft
- 710 2_ |a Institute of Electrical and Electronics Engineers.
- 710 2_ |a IEEE Communications Society.
- 776 08 |i Online version: |t IEEE journal on selected areas in communications (Online) |
x 1558-0008 |w (DLC) 2005215206 |w (OCoLC)44607980
- 776 08 |i Online version: |t IEEE journal on selected areas in communications |
w (OCoLC)752408668
- 850 — |a AAP |a AU |a ArU |a AzU |a CCC |a CLSU |a CPT |a CSt |a CU |a CU-A |a CU-I |a CU-RivA |a CU-S |a CU-SB |a CaAEU |a CaBVaS |a CaMWU |a CaNSHT |a CaOTDT |
a CaOTU |a CaOWA |a CaOWU |a CaQMH |a CaSSU |a CoFS |a CoU |a CtY-KS |
a DGW |a DLC |a DeU |a FU |a GAT |a ICJ |a ICU |a ICarbS |a IEN |a IU |a IU |a IaAS |
a IaU |a InLP |a KMK |a KyU |a LNT |a LU
- 850 — |a MBU |a MCM |a MMeT |a MdU |a MiD |a MiDW |a MiEM |a MiU |a MnU |a MoKL |
a MoKU |a MoSW |a MoU |a MsSM |a MsU |a MtBC |a N |a NN |a NNC |a NNE |a NRU |
a NSyU |a NcD |a NcRS |a NcU |a NdU |a NhU |a NjP |a NjR |a NmU |a OCU |a OU |
a OkU |a OrCS |a PBL |a PPD |a PPiC |a PPiU |a PU |a RPB-S |a ScCleU |a ScU |
a TxArU |a TxCM |a TxDaM |a TxHR |a TxU |a ULA |a ViBibV |a ViU |a ViU |a WM |
a WaU |a WwU |a WyU
- 859 41 |u <http://ieeexplore.ieee.org/servlet/opac?punumber=49>
- 890 — |a IEEE journal on selected areas in communications. New York, N.Y. 1983- . |
i 83-643067

906	—	a 7 b cbc c serials d 2 e ncip f 19 g n-oclcserc
920	—	a Keep 1
923	—	d 20020822 s Group Registration
984	—	a srvf
985	—	e eserial 200406
991	—	b c-GenColl h TK5101.A1 i I35 w SERIALS
991	—	b c-GenColl h TK5101.A1 i I35 p 00002542122 t Copy 1 v 5 * 1987 w CCF
991	—	b c-GenColl h TK5101.A1 i I35 p 00014156616 t Copy 1 v 5 * 1987 Jan-June w CCF
991	—	b c-GenColl h TK5101.A1 i I35 p 00014155739 t Copy 1 v 6 * 1988 w CCF
991	—	b c-GenColl h TK5101.A1 i I35 p 00002542158 t Copy 1 v 6 Jan-June * 1988 w CCF
991	—	b c-GenColl h TK5101.A1 i I35 p 00014155727 t Copy 1 v 7 Jan-June * 1989 w CCF
991	—	b c-GenColl h TK5101.A1 i I35 p 00034557743 t Copy 1 v 1 w CCF
991	—	b c-GenColl h TK5101.A1 i I35 p 0001415632A t Copy 1 v 3 w CCF
991	—	b c-GenColl h TK5101.A1 i I35 p 00014155715 t Copy 1 v 7 w CCF
992	—	b SER w SERLOC

[Request this Item](#)
[LC Find It](#)

Item Availability

[>](#)

CALL NUMBER	TK5101.A1 I35 Set 1
Request in	Jefferson or Adams Building Reading Rooms
Status	c.1 v. 41, no. 5 2023 May At Bindery - 01-17-2025
Older receipts	v.1 (1983), v.4:no.1-v.9:no.1 (1986:Jan.-1991:Jan.), v.9:no.4-v.14:no.6 (1991:May-1996:June), v.14:no.9-v.18:no.1 (1996:Sept.-2000:Jan.), v.18:no.3-v.19:no.9 (2000:Mar.-2001:Sept.) v.19:no.11-v.22:no.5 (2001:Nov.-2004:June), v.23:no.1-v.23:no.3 (2005:Jan.-2005:Mar.), v.23:no.5-v.25:no.6 (2005:May-2007:Aug.), v.25:no.8-v.25:no.9 (2007:Oct.-2007:Dec.), v.31:no.8-v.41:no.3 (2013:July-2023:Mar.) v.41:no.5-v.41:no.12 (2023:May-2023:Dec.)
Supplements	v.24:no.8 (2006:Aug.), v.24:no.12 (2006:Dec.), v.25:no.4 (2007:Apr.),

v.25:no.6 (2007:Aug.), v.25:no.9 (2007:Dec.), v.26:no.6-v.26:no.1-
v.26:no.2 (2008:Jan.-2008:Feb.), v.26:no.9-v.31:no.6
(2008:Aug.-2013:June), v.31:no.9 (2013:Sept.)

CALL NUMBER	TK5101.A1 I35 Copy 1 Unbound issues
Request in	Newspaper & Current Periodical Reading Room (Madison LM133)
Latest receipts	v. 42, no. 12 (2024 Dec.) v. 42, no. 11 (2024 Nov.) v. 42, no. 10 (2024 Oct.) v. 42, no. 9 (2024 Sept.) v. 42, no. 8 (2024 Aug.) v. 42, no. 7 (2024 July) v. 42, no. 6 (2024 June) v. 42, no. 5 (2024 May) v. 42, no. 4 (2024 Apr.) v. 42, no. 3 (2024 Mar.) v. 42, no. 2 (2024 Feb.) v. 42, no. 1 (2024 Jan.) v. 26, no. 5 (2008 June)

Ex. 1019C

Ex. 1019C
Citations to Hui

Source Document:

Hui, J. Y., Karasan, E., Li, J., & Zhang, J. (1996, January). Client-server synchronization and buffering for variable rate multimedia retrievals. *IEEE Journal on Selected Areas in Communications*, 14(1), 226-237.

Early Citations: 1996 – 1997 (in chronological order)

Huang, C. M., & Lee, R. Y. (1996, July). Quantification of quality-of-presentations (QOPs) for multimedia synchronization schemes. *ACM SIGCOMM Computer Communication Review*, 26(3), 76-104.

Lee, Y. B., & Wong, P. C. (1996, October). Designing a Server Array System for Multimedia World-Wide-Web Services. In *Workshop on Resource Management in Computer Systems and Networks, Eighth IEEE Symposium on Parallel and Distributed Processing, New Orleans, Louisiana*.

Daami, M., & Georganas, N. D. (1997, June). Client based synchronization control of coded data streams. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems* (pp. 387-394). IEEE.

Kaladji, F., Ishibashi, Y., & Tasaka, S. (1997, September). Performance evaluation of a dynamic video resolution control scheme in media-synchronized video and audio communications. In *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications (Cat. (Vol. 3, pp. 1642-1646)*. IEEE.

DIGITAL VIDEO COMPRESSION

- Compression fundamentals
- MPEG-2 and MPEG-4
- H.264 codec
- Much more!

FEATURING:

JVT / H.264 / MPEG-4 Part 10-
THE NEW COMPRESSION
STANDARD



PETER SYMES

CD
INCLUDED



Google Exhibit No. 1020
Page 001

Google Exhibit No. 1063
Page 0691

Digital Video Compression

Peter Symes

McGraw-Hill

New York Chicago San Francisco Lisbon London Madrid
Mexico City Milan New Delhi San Juan Seoul
Singapore Sydney Toronto

Humber College Library
205 Humber College Blvd
Toronto, ON M9W 5L7

Google Exhibit No. 1020
Page 002

Google Exhibit No. 1063
Page 0692

Cataloging-in-Publication Data is on file with the Library of Congress

Copyright © 2004, 2001 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 0 9 8 7 6 5 4 3

P/N 142494-6
PART OF
ISBN 0-07-142487-3

The sponsoring editor for this book was Stephen S. Chapman and the production supervisor was Sherri Souffrance. It was set in Vendome by Patricia Wallenburg. The art director for the cover was Anthony Landi.

Printed and bound by RR Donnelley.

Portions of this book were previously published in *Video Compression Demystified*, © 2001.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to the Director of Special Sales, McGraw-Hill Professional, Two Penn Plaza, New York, NY 10121-2298. Or contact your local bookstore.



This book is printed on recycled, acid-free paper containing a minimum of 50% recycled, de-inked fiber.

Information contained in this work has been obtained by The McGraw-Hill Companies, Inc. ("McGraw-Hill") from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantee the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

CONTENTS

Preface	xv
Acknowledgments	xvii
1 What Is Compression?	1
Introduction	2
Information and Data	2
Information Reduction	4
Lossless Compression	5
Run-Length Encoding	5
Entropy Encoding	6
Lossy Compression	8
Image Compression Standards	10
Symmetric and Asymmetric Systems	11
Why Do I Care?	12
Upcoming Topics	12
2 An Introduction to Images	15
Introduction	16
Video or Computer Graphics?	16
Sampling and Quantization	17
Sampling Rate and Aliasing	18
One-Dimensional Sampling	18
Two-Dimensional Sampling	26
Temporal Sampling	29
3 Entropy Coding	35
Introduction	36
Entropy	36
Huffman Codes	41
Dangers of Variable-length Coding	43
Modified Huffman Codes	44
Arithmetic Encoding	47

4	Predictive Coding	49
	Markov Sources	50
	Differential Pulse Code Modulation	54
	Predicting Image Values	56
	Adaptive Prediction	59
5	Transforms	67
	Introduction	68
	Time, Space, and Frequency Domains	70
	Frequency and Spatial Frequency	72
	The Discrete Cosine Transform	74
	The Fourier Approach to DCT	76
	DCT in Terms of Basis Functions	81
	DCT as Axis Rotation	85
	DCT Examples	87
	DCT Failure	88
6	Quantization	91
	Introduction	92
	Mean Square Error	92
	Types of Quantizer	93
	Uniform Scalar Quantizer	94
	Nonuniform Scalar Quantizers	95
	The Lloyd-Max Quantizer	97
	Entropy-constrained Quantizers	97
	Vector Quantization	98
	Applications of Quantization	100
	Direct Image Quantization	100
	Quantization with Predictive Coding	103
7	JPEG	107
	Introduction	108
	Baseline JPEG	109
	Image Data Encoding	109
	JPEG Baseline Encoding and Decoding	110
	DCT Transform	113
	Quantization	116
	Scanning, Descriptors, and Entropy Coding	119

Assembling the Bit Stream	125
Parsing the Received Bit Stream	125
Recovering the Quantized Coefficients	126
Dequantization	127
Inverse DCT	127
Comparison	129
Examples of Baseline JPEG	129
JPEG Extensions	130
Motion JPEG	131
8 Motion Compensation	137
Introduction	138
Temporal Redundancy	138
Motion Aliasing	138
The Motion Compensation Approach	139
Static Backgrounds	139
Motion Vectors	141
Block Matching	143
Matching Criteria	143
Optimum Block Size	143
Full-Search Block Matching	145
Hierarchical Block Matching	146
Residuals	147
Other Motion Estimators	148
Restricted Search Systems	148
Phase Correlation Motion Estimation	149
9 MPEG-1	151
Introduction	152
What MPEG Defines	153
Hierarchy and Terminology	154
Frame Types in MPEG	156
Intraframes (I-frames)	156
Non-intra Frames (P-frames and B-frames)	156
MPEG Group of Pictures	158
Coding of Macroblocks	160
Intracoding of Macroblocks	161
Non-intracoding of Macroblocks	162

Rate Control	165
The MPEG Encoder and Decoder	169
10 MPEG-2	171
Introduction	172
MPEG-2 Enhancements	173
Color Space	173
Slice Structure	173
Quantization	173
Concealment Motion Vectors	174
3:2 Pulldown	175
Pan and Scan	175
MPEG-2 Profiles and Levels	175
Interlace Tools	178
Frame and Field Pictures	179
Frame and Field DCT	180
Frame and Field Prediction	182
Scalable Coding Profiles	183
MPEG-2 System Layer	184
Packetized Elementary Stream	185
Program Stream	185
Transport Stream	186
Practicing the Art of MPEG	189
Contributors to Poor Performance	189
MPEG Artifacts	190
Tips for Higher Quality	191
Update 2003	192
11 MPEG-4	195
Introduction	196
Video in MPEG-4	200
MPEG-4 Video Hierarchy	200
Shape Coding	201
Texture Coding	203
Boundary Coding	205
Coding of Arbitrary-Shaped Video Objects	205
Sprites	206
Static Texture Coding	208

Animations	208
Scalability	209
Advanced Coding Extensions (ACE)	211
Visual Profiles	212
Scene Compositing and Interaction	216
Scene Modeling	216
Interaction	217
Work in Progress	217
Studio Profiles	217
Fine Grain Scalability	218
Conclusion	220
Update 2003	221
12 Joint Video Team (JVT)	223
Introduction	224
JVT Requirements	226
A Summary of the JVT Codec	229
Similarities of JVT to MPEG-2	230
New Concepts and Technologies	230
JVT Tools	232
Transform	232
Logarithmic Quantization	235
Spatial Prediction	236
Deblocking Filter	236
Motion Estimation	238
Temporal Prediction	240
Macroblock-adaptive Frame/Field Coding	242
Lossless Coding	242
Profiles and Levels in JVT	243
Baseline Profile	243
Main Profile	244
Extended Profile	244
Levels	244
Future Work of JVT	245
13 Windows Media	247
Introduction	248
Windows Media History	248

Windows Media 9 Codecs	249
Windows Media Audio 9	250
Windows Media Audio 9 Professional	250
Windows Media Audio 9 Voice	250
Windows Media Audio 9 Professional Lossless	251
Windows Media Video 9 (WMV-9)	251
Windows Media Video 9 Screen	251
Windows Media Video 9 Image	251
Windows Media Video 9	252
Frame Interpolation in WMV	254
Windows Media 9 Licensing	256
14 Intellectual Property and Licensing	257
Introduction	258
Standards Bodies and Patents	258
Licensing Bodies	259
Windows Media 9 Licensing	261
JVT Licensing	262
15 MPEG-7 and MPEG-21	267
Introduction	268
MPEG-7	268
Concepts of MPEG-7	268
MPEG-7 Terminology	271
MPEG-7 Structure	272
MPEG-7 Visual	273
Summary	274
MPEG-21	276
16 Pro-MPEG and MPEG Operating Ranges	279
Introduction	280
Goals of Pro-MPEG	281
Update 2003	285
17 DV Compression	287
Introduction	288
Basic Concepts of DV Compression	289
Detailed Description	290

25 Mbits/s Compression	290
50 Mbits/s Compression	298
Update 2003	299
18 Wavelets	301
Introduction	302
More about Fourier Transforms	302
Wavelets Concept	306
Wavelets as Filters	307
Wavelet Compression	312
19 JPEG2000	317
Introduction	318
Limitations of the Original JPEG System	318
Goals of JPEG2000	319
20 Audio Compression	321
Introduction	322
Masking in Human Hearing	323
Simple Audio Compression Schemes	324
Quadrature Mirror Filters	326
Subband Coding	330
Bit Allocation	332
Transform Coding	332
Example Compression Systems	333
Audio Compression in MPEG	333
Audio Compression for ATSC	334
21 Streaming Media	339
Introduction	340
Applications for Streaming Media	341
Standards for Streaming Media	343
Update 2003	345
22 Closing Thoughts	347
Introduction	348
Fractal Compression	348
Statistical Multiplexing	349

Concatenated Compression Systems	351
Switching MPEG	354
MPEG Applications	358
Some Solutions	359
Mezzanine Compression Systems	361
A Glimpse into the Future	364
A Glossary	365
B Bibliography	377
C Internet Resources	381
D About the CD-ROM	385
Index	387

CHAPTER **1**

What Is Compression?

Introduction

Compression is the science of reducing the amount of data used to convey information. In this book we will examine a wide range of techniques—some simple, some very complex—and ways in which these techniques may be applied to video.

Compression relies on the fact that information, by its very nature, is not random but exhibits order and patterning. If this order and patterning can be extracted, the essence of the information can often be represented and transmitted using less data than would be needed for the original. We can then reconstruct the original, or a close approximation of it, at the receiving point.

There are several families of compression techniques, fundamentally different in their approach to the problem. These techniques are in fact so different that they can often be used sequentially to good advantage. Sophisticated compression systems use one or more techniques from each family to achieve the greatest possible reduction in data.

There are arguments about the semantics of compression. Some authorities use the term *compression* only when there is some resultant impairment, and they prefer terms like *bit rate reduction* for techniques that involve no loss. In this book *compression* is used to cover all techniques that reduce data, whatever the nature of the process and whether or not errors are introduced.

Before examining compression techniques, let us look at the general issue of information and the data used to represent it.

Information and Data

A major concern in today's world is the handling of information. The information may be of many types—written text, the spoken word, music, still pictures, and moving pictures are just a few examples. Whatever the type of information, we can represent it by electrical signals or data, and transmit it or store it.

The more complex the information, the more data is needed to represent it. Plain text can be represented by 8 bits/character, or about 20 kbits for a page. CD-quality music requires nearly 1,500 kbits for each second, and full-motion 525-line video (as used in North American television systems) needs about 200 Mbits (200,000 kbits) for each second.

When we need to transmit a given amount of data, we use a certain bandwidth (measured in bits per second) for the time necessary to transmit all the bits. In the real world, there is always some restriction on the available bandwidth. For example, if the transmission must be over regular telephone lines using modems, it is difficult to achieve more than 50 kbits/s (kilobits per second). This is fast compared to the 300-bits/s modems of a few years ago, but even at 50 kbits/s it would take some two hours to transmit a single second of high quality video.

Sometimes it is a practical restriction, like the use of a modem, that sets the requirements for compression. We could send high-quality uncompressed audio data over a modem, but each second's worth of audio would take about 30 seconds to transmit. In other words, we would have to receive the data gradually, store it away, then play the resulting file at the correct rate to hear the sound. However, we may want to transmit sound so that we can listen to it as it arrives in "real time." We could choose to reduce the quality we are trying to send, but even telephone-quality speech needs a data rate substantially higher than our high-speed modem can provide. If we want to send real-time audio over a modem link, we have to compress the data.

Another example of this type of requirement is the digital television system developed for North America. When the study was started in 1977, few thought it would be possible to transmit high-definition television (characterized by the 1,125-line system) over a single 6-MHz channel as used for today's 525-line television. When digital techniques were considered, the problem looked even worse, because the high-definition signal represented over 1 Gbits/s of data. In the event, two teams of engineers combined to produce the answer: The transmission engineers designed a system to deliver nearly 20 Mbits/s reliably over a 6-MHz channel, and the compression engineers achieved data reduction of about 60 to 1, while maintaining excellent picture quality.

At other times the motive for compression is financial. Bandwidth costs money, and so compression reduces the cost of transmission. A satellite link with a data bandwidth of 40 Mbits/s will earn more money if it can carry ten television channels instead of just one. I may want to download a 6-MB file from a computer online service; if a compressed version is available that is only 1.2 MB, I will save 80 percent of the online charges.

Transmission and storage of information might seem to be very different problems but, in terms of quantity of data, the economics are the same. Just as there is a cost for each bit transmitted, there is a

cost for each bit stored in memory, on disk, or on tape. If we can use less data, both transmission and storage will be cheaper. There are some differences in how compressed data is best structured, and these will be mentioned in Chapter 10. Otherwise, everything we say about compression applies equally to transmission or storage.

The motives for wanting to reduce the amount of data should be clear. Now we will examine some of the techniques.

Information Reduction

One important step in compression technology is easy to ignore because it lacks the glamor of some of the sophisticated techniques. It is extremely important to ensure that the signal or data stream we want to compress represents the information we want to transmit, and *only* the information we want to transmit. If the data stream contains surplus information of any nature, this information will take bits to transmit and result in fewer bits being available for the information we do need. Surplus information is *irrelevant* because the intended recipient can make no use of it.

Surplus information can take many forms. It can be information in the original signal or data stream that exceeds the capabilities of the receiving device. For example, there is no point in transmitting more resolution than the receiving device can use, and no point in transmitting color information if the receiver is known to be monochrome.

Another form of surplus information is *artifacts*—features or elements of the input that are not truly part of the information. *Noise* is the most obvious example; almost anything derived from a physical transducer has some component of noise. Noise is by nature random or nearly so and, as we shall see, this makes it essentially incompressible. In video, noise may come from many sources including the transducer, analog processing, and film grain. When the signal has been converted to digital form, it may have additional *quantization noise*, and digital processing may add noise through truncation.

Many other types of artifacts exist, ranging from filter ringing to film scratches. Some may seem trivial, but in the field of compression they can be very important. Compression relies on order and self-consistency in a signal. An artifact—anything that does not “belong”—tends to require a disproportionately high number of bits to transmit, just because it destroys this order and self-consistency.

This book concentrates on the techniques of compression. Preprocessing of video to remove irrelevant information and artifacts is a very complex subject in itself. A preprocessing unit developed by Snell & Wilcox includes no less than seven different filters!

Lossless Compression

Some compression techniques are truly lossless. In other words, when we have compressed some data, we can reverse the process (*decompress*) and get the same data we started with, exactly and precisely. Lossless compression works by removing *redundant* information—information that, if removed, can be recreated from the remaining data.

In one sense, this is the ideal of compression; there is no cost to using it other than the cost of the compression and decompression processes. Unfortunately, lossless compression suffers from two significant disadvantages. Typically, lossless compression offers only relatively small compression ratios, so used alone it often does not meet economic needs. Also, the compression ratio is very dependent on the input data. Used alone, lossless compression cannot guarantee a constant output data rate such as might be required for a transmission channel.

For some applications, lossless is the only type of compression that can be used. If we wish to transmit a binary file such as a computer program, receiving an approximation to the original is of no use whatsoever. The program will run only if replicated exactly. Fortunately, many computer files have a high degree of order and patterning, so lossless compression techniques do yield useful results. Compression programs like the well-known PKZIP use a combination of lossless techniques and are particularly effective on graphics image files.

One advantage of lossless compression is that it can be applied to any data stream. Most video compression schemes use lossy techniques to achieve large degrees of compression. Lossless techniques are then applied to the resulting data stream to reduce the data rate even further. Two lossless techniques are in general use.

Run-Length Encoding

Many data streams contain long runs of a value. For example, a graphics file will typically contain values for each pixel, ordered line by

line. A red object will likely be represented by several long runs of the value "red." With appropriate coding it can use much less data to say "red 23 times" instead of "R R R R R R R R R R R R R R R R R R R R R R R." This is run-length encoding—we code the length of the "run" of a certain value. In video compression, various tricks can be played to make the data suitable for run-length encoding.

Entropy Encoding

Entropy encoding is usually the last step in compression. Whereas run-length encoding relies on adjacent values being the same, entropy encoding looks at the overall frequency of specific values, wherever they are located. The implementation of entropy encoding is quite complex, but the principle is easy to understand if we define our terms carefully.

If the data is represented by bytes (groups of 8 bits), each byte can have any *value* between 0 and 255_{10} . When we come to transmit this data we will choose a *symbol* to represent each possible value, and we transmit a sequence of *symbols* until all the *values* have been sent.

The simplest way to code the data is to send the bytes themselves; in other words, each symbol is the same as the value it represents. This is so obvious that we tend not to think of anything else, and make no distinction between the value and the symbol. But it does not have to be like this. So long as both the transmitting and receiving points know and apply the same set of rules, we can use any set of 256 unique symbols to represent the 256 possible values. This is not very helpful until we add another factor: Provided we have a set of rules that lets us know when one symbol ends and another begins, *not all the symbols need to be the same length*.

This is the essence of entropy encoding. It compresses the data by using short symbols to represent values that occur frequently, and longer symbols to represent the values that occur less frequently. Unless the input data is very close to random in its distribution of values, this technique means that total data used for the symbols will be less than the total of the original values.

Optimization of entropy encoding is complex, but the basic idea is neither difficult nor new. Let us compare two ways of coding the alphabet. Anyone who has spent much time with computers has come across the ASCII encoding system. In ASCII we represent any letter by a particular set of 8 bits, or 1 byte. It does not matter what *value* (letter)

we choose, the *symbol* is 8 bits long.¹ A much earlier method of encoding the alphabet was invented by Samuel Morse in 1838. The Morse code is a simple but useful example of an entropy coding system.

The Morse code uses combinations of dots and dashes as symbols to represent letters. A dash is 3 times as long as a dot; each dot or dash within a symbol is separated by one dot length; each symbol is separated by one dash length. Morse designed his code to transmit English-language text, and to maximize the speed of transmission he looked at the frequency with which the various letters of the alphabet appear in typical text. The letter *e* is the most common letter in English, so Morse assigned the shortest possible symbol—one dot (actually four dots in length because of the intersymbol space). The next most common letter *t* received the next shortest symbol, one dash. At the other extreme, an infrequent letter like *y* receives three dashes and a dot, a symbol four times as long as that used for *e*, as shown in Figure 1-1.

Figure 1-1
Morse Code provides a simple example of variable length encoding.

e		4
t		6
c		14
y		16
z		14

The Morse code also illustrates one of the problems of entropy encoding. For transmission to be successful, obviously both the transmitter and the receiver must use the same mapping of symbols to values. The relationship between values and symbols is known as the *code*

¹ Of course, ASCII encodes numbers as well as letters; it encodes uppercase and lowercase letters, and it encodes punctuation and other symbols. If we need to encode the letters only, without regard to case, we do not need 8 bits because there are not 256 letters. The old telex coding using 5-bit symbols would be sufficient. However, the comparison of fixed and variable symbol lengths is still valid.

table. But the most efficient mapping can be determined only when the frequency of values is known. The code table for Morse is derived from the frequency of letters in English text. If we wanted to send Polish-language text, where the letters *c*, *y*, and *z* are used more frequently than in English, this code table would not be very efficient because those letters have long symbols. If we wanted to transmit a few lines of Polish in the middle of many pages of English, this is probably not a major concern. However, if we had to transmit a very large volume of Polish, we might want to change the code table. We can do this if we have established rules at the transmitter and receiver to allow for different code tables, but we would need to evaluate this decision carefully. If we decide to change code tables, we have to transmit the new table, and that means transmitting a considerable amount of additional data!

There is no perfect solution to this, and the best compromise depends on the application. Some coding schemes use only one code table; this may not always be the most efficient table, but the coding system does not carry any of the overhead necessary to change tables. Other systems have a small number of fixed code tables, known to both transmitter and receiver. The encoder can examine the data to be transmitted, and select the most efficient code table. The only overhead is the provision of a message that tells the receiver "Use table number 3." Still other schemes allow for the transmission of custom code tables whenever the characteristics of the data change substantially.

Lossy Compression

Lossless compression would be an ideal answer, except that it rarely provides large degrees of compression, and it cannot be used alone to guarantee a fixed bit rate. Lossless compression is an important part of our tool kit, but on its own it does not provide a solution to many practical problems. If we want to put digital audio over a modem link, or high-definition television through a 6-MHz channel, we have to accept that the compression process will result in some loss; what we get out will not be exactly the same as what we put in. This is the field of *lossy compression*. Ideally, lossy compression, like information reduction, removes *irrelevant* information. Some information is truly irrelevant in that the intended recipient cannot perceive that it is missing. In most cases we also look for information that is close to irrelevant, where the quality loss is small compared to the data savings.

The objective of lossy compression is simple. We want to get maximum benefit (compression ratio, or bit rate reduction) at minimum cost (the loss in quality). However, the realization of this objective is not simple. As we will see when we look in detail at lossy compression techniques, a very large number of parameters have to be chosen for any given implementation, and many of these parameters must be varied according to the dynamic characteristics of the data. Even with a simple measure of the resulting quality (or lack of it), optimizing such a large number of variables is a complex task. Because these schemes rely on a knowledge of how the information will be perceived by the recipient, they are known as *perceptual coding systems*.

Unfortunately, there is as yet no simple measure that can be applied to the quality of a compression system, particularly for video. The only measure that really matters is the subjective effect as perceived by a viewer, and this is an extremely complex function. If compression loss were confined to one characteristic of a picture, it would be relatively simple to derive a way of measuring this characteristic objectively. We could then perform careful subjective tests on a representative sample of viewers and arrive at a calibrated relationship between this characteristic and subjective quality. Unfortunately, compression loss results in not one but many changes to the characteristics of the picture; each of these has a complex, nonlinear relationship to subjective quality, and they interact with each other! (As an indication of the difficulty of quantifying compression loss in images: Workers in the field consistently observe that the perceived quality of a picture improves as the quality of the accompanying sound improves.)

Compression loss in images involves two main components: things that should be in the picture, but are lost, and things that are added to the picture (artifacts) that should not be there. Areas where we would expect loss include spatial resolution (luminance and color, probably to different degrees) and shadow and highlight detail. Where extreme compression is needed for applications like video conferencing, or the Internet, temporal resolution would also be sacrificed. Artifacts that might be added include blocking, "mosquito noise" in edges, quantization noise, stepping of gray scales, patterning, ringing, and the like. As a further complication, in the presence of loss—say of resolution—the addition of small amounts of artifact such as noise or ringing may actually improve the subjective quality of the picture.

Even within "standardized" compression schemes such as MPEG, the tuning of a system to yield the maximum quality per bit is a black art involving models of the human psychovisual system that are closely guarded commercial secrets. Evaluation of the effect of parameter changes is very complex for the reasons discussed above.

There may be light at the end of the tunnel. In the fall of 1997, Tektronix released a product called the *signal quality analyzer*. This device uses a technology known as *just noticeable difference* (JND) developed jointly by Tektronix and Sarnoff Corporation. It compares an output sequence from a compression system with the known input sequence, and weights each error according to its type, severity, location, and the masking effect of other nearby picture elements. The intent is to model the behavior of the human eye-brain combination, and initial tests show an encouraging degree of correlation between the measurements made by the system and subjective testing with observers.

Image Compression Standards

Any practical compression system requires the use of a number of compression tools. These tools are combined to provide a cost-effective and optimized solution for the particular problem at hand. In proprietary systems where compression is used only internally, the designer has a free hand in selecting what tools will be used, and how. In a professional videotape machine such as the Sony Digital BetaCam®, the compressed data is used only within the machine. All connections to the outside world are made as standard uncompressed video, so that there is no need for anyone outside the manufacturing organization to know the details of the compression scheme.

This approach works as long as the compressed data stays within one manufacturer's system. We may wish to send compressed data over a network or a broadcast channel, or place it on a CD, and have it usable by a different manufacturer's equipment at the other end. In this case, everyone needs to know how to decompress the data, even if they do not know exactly how the compression was achieved. In other words, we need one or more standardized compression systems.

In the field of electronic imagery, standardization work has been undertaken by the International Organization for Standardization (ISO) in cooperation with the International Telecommunications Union and the International Electrotechnical Commission (IEC). The

Joint Photographic Experts' Group and the Motion Picture Experts' Group produced the well-known JPEG, MPEG-1, and MPEG-2 standards that form the basis of most image compression today. In 1996, the National Academy of Television Arts and Sciences recognized the importance of this work by the award of an Emmy statue to ISO and IEC.

The JPEG and MPEG standards, and the continuing work of the MPEG committee, are discussed in later chapters.

Symmetric and Asymmetric Systems

Compression systems are, in general, complex and quite expensive. One characteristic important in the choice of a system is the degree of *symmetry* or *asymmetry*. This is best illustrated by examining two different scenarios.

Consider the application of videoconferencing, a subject of great interest to compression specialists. To gain popularity for other than special events, videoconferencing must be inexpensive and convenient. To be inexpensive, it must use a minimum of bandwidth, and the transmitting and receiving equipment must be affordable. To achieve low bandwidth, we need compression, and every conference station needs the ability to compress and decompress. This is a symmetrical situation—there is no benefit in designing a system where compression is more or less complex and expensive than decompression.

In contrast, some applications, such as television broadcasting, are characterized by a single compressor feeding a very large number of decompressors. CD-ROMs and digital video disks (DVDs) have similar economics; one compression system is used to produce the master, but millions of players decompress the data. These are asymmetric models. If we can reduce the data rate, or simplify the decompressor, by additional complexity in the compressor, the system as a whole benefits economically.

Baseline JPEG is very close to a symmetric system. Encoder and decoder complexities are very similar. In contrast, MPEG, with its use of motion compensation, is an asymmetric system. The bit rate is reduced with relatively little increase in decoder complexity, but at the expense of much greater encoder complexity and cost.

Why Do I Care?

It is tempting for the television engineer to conclude that compression is a subject for experts, and that it takes place within a black box. In many ways this is valid; an engineer designing or operating a television facility does not need to be able to design a television camera, or a routing switcher, and the same is true for compression systems. However, good engineers understand in principle what is happening in each of the major components of a system, even if they could not design that component.

Compression will exist at some point in virtually all television systems. The performance of the compression is affected, sometimes dramatically, by characteristics of the video signal. A signal that has been compressed carries a signature, often invisible, that can radically alter the behavior of downstream operations.

I believe that video engineers must understand the processes that are being applied to their images. Only in this way can they protect quality and diagnose problems. Anyway, compression is a fun subject!

Upcoming Topics

In Chapter 2 we look at many of the fundamentals of the sampled images that form the actual input to any compression system. Many of the topics covered are not specific to compression, but all are fundamental to its operation. Most engineers will be familiar with this material, but I encourage you to read the chapter, anyway. These ideas are absolutely essential to the operation of compression systems, and many so-called experts go wrong because they fail to remember or apply the fundamentals discussed here.

Chapters 3 to 6 cover the basic tools of compression systems for static images, and Chapter 7 brings these tools together in a description of the JPEG system. Chapter 8 introduces the special characteristics of moving images and the technology of motion compensation. Having dealt with that, we can then move on, in Chapters 9 and 10, to MPEG-1 and MPEG-2.

Digital Video Compression concentrates largely on JPEG and MPEG because these are still the most important systems for the video engineer today. Although there are more modern techniques, the large number of MPEG-2 decoders deployed in digital TV receivers (includ-

ing satellite receivers) and in DVD players is sufficient to ensure that this standard will be important for many years. MPEG-4, in its early form, is described in Chapter 11. The latest work of the standards bodies, and the latest part of MPEG-4, comes from the Joint Video Team (JVT) of MPEG and ITU-T, and is discussed in Chapter 12. Another new contender for attention is Windows Media 9, covered briefly in Chapter 13. Chapter 17 describes the DV compression system, which uses similar technology to MPEG, but in a quite different way. Later chapters explore the principles and application of other technologies, including wavelets and fractals, and the fundamentals of audio compression.

No book on compression would be complete without a look at streaming video and audio on the Internet, and this leads to a description of the contents of the CD-ROM included with this book. Here you will have the opportunity to explore some of the concepts directly through demonstration programs and sample images and sequences.

The book is organized in a progressive manner—each chapter assumes knowledge of at least the basic elements of previous chapters—so it is a good idea to work through the book in the correct sequence. However, each chapter covers the basics first, and then goes into greater detail. The fundamentals are important, and much of the detail is provided for example, and because I find it fascinating. If the detail gets too much, move on to the next chapter—I had fun writing this book, and I want you to enjoy reading it!

CHAPTER **11**

MPEG-4

Introduction

The wheels of international standardization grind slowly, and to ensure that a standard is eventually achieved there are strict rules that prohibit substantive change after a certain point in the process. By the time a standard is officially adopted there is often a backlog of desired enhancements and extensions. So it was with MPEG-2. As discussed in Chapter 10, MPEG-3 had been started and abandoned, so the next project became MPEG-4.

MPEG-4 exists in two parts, Version 1 and Version 2, which is a superset of Version 1. In other words, all Version-1 constructs are valid within Version 2. Work began in 1993, and was formalized in July 1995 with a call for proposals. The various proposals for audio and video were evaluated by subjective testing and by experts. Drafting of the Version 1¹ followed, leading to a Committee Draft in November 1997, and became an ISO/IEC International Standard in 1999. Meanwhile, work continued on the Version-2 extensions and the Draft Amendment was frozen in December 1999, with various phases of formal adoption during 2000. As of the summer of 2000, work on extensions continues forming the basis for Amendments 2 and 3, representing Versions 3 and 4 respectively. Even more work is planned, to determine and meet the needs of digital cinema.

At first, the main focus of MPEG-4 was the encoding of video and audio at very low rates. In fact, the standard was explicitly optimized for three bit rate ranges:

- Below 64 kbits/s
- 64 to 384 kbits/s
- 384 kbits/s to 4 Mbits/s

Performance at low bit rates remained a major objective and, as we shall see, some very creative ideas contributed to this end. Great attention was also paid to error resilience, making MPEG-4 particularly suitable for use in error-prone environments, such as transmission to personal handheld devices. Other profiles and levels use bit rates up to 38.4 Mbits/s, and work proceeds on studio-quality profiles and levels using data rates up to 1.2 Gbits/s.

More important, MPEG-4 became vastly more than just another compression system—it evolved into a totally new concept of multi-

¹ Except for Conformance Testing, which followed about a year later.

media encoding with powerful tools for interactivity and a vast range of applications. Even the official "overview" of this Standard spans 67 pages, so in this chapter I can only attempt a brief introduction to the system. I hope I will manage to convey some of the new concepts and demonstrate the enormous richness of MPEG-4; it really is a Standard with something for everyone, and each facet of the Standard offers a wealth of tools and options. Despite the focus on the low bit rate applications, MPEG-4 includes important extensions for high-end studio operations and high-definition television.

The most significant departure from conventional transmission systems is the concept of *objects*. Different parts of the final scene can be coded and transmitted separately as *video objects* and *audio objects* to be brought together, or *composited*, by the decoder. This raises a number of interesting possibilities. Most obviously, different types of objects may be coded using different techniques, and each may be coded with the tools most appropriate to the job. The different objects may be generated independently, or in some cases a scene may be analyzed to separate, for example, foreground and background objects. An interesting example was demonstrated recently. Video coverage of a soccer game was processed to separate the ball from the rest of the scene. The background (the scene without the ball) was transmitted as a "teaser" to attract a pay-per-view audience. Anyone could see the game, but only those who paid could see the ball! Figure 11-1 shows the general concept.

As we have seen throughout this book, no coding system is universally efficient. DCT and quantization are effective for continuous-tone, bandwidth-limited images, but certainly not optimal for everything that we may wish to display. As a simple example, if a scene is to be overlaid with text, conventional systems key the text into the video, often filling the body and edges of the letters with contrasting colors. The resulting combination is then coded by normal means. Superimposing text in this manner adds a good deal of high-frequency energy to the signal, and these edges are not well coded by DCT. They consume a disproportionate number of bits, and the result is often less than pleasing. One of the very noticeable artifacts of DCT-based encoding is the effect known as *mosquito noise* on sharp edges such as overlaid text, caused by excessive quantization of the high-frequency coefficients.

Thus, the addition of text has a significant impact on the efficiency of the video coding. Yet we know that we could code that text in a very simple manner—the letters can be specified in ASCII (or some more efficient code), and position, font, size, color, and spacing information

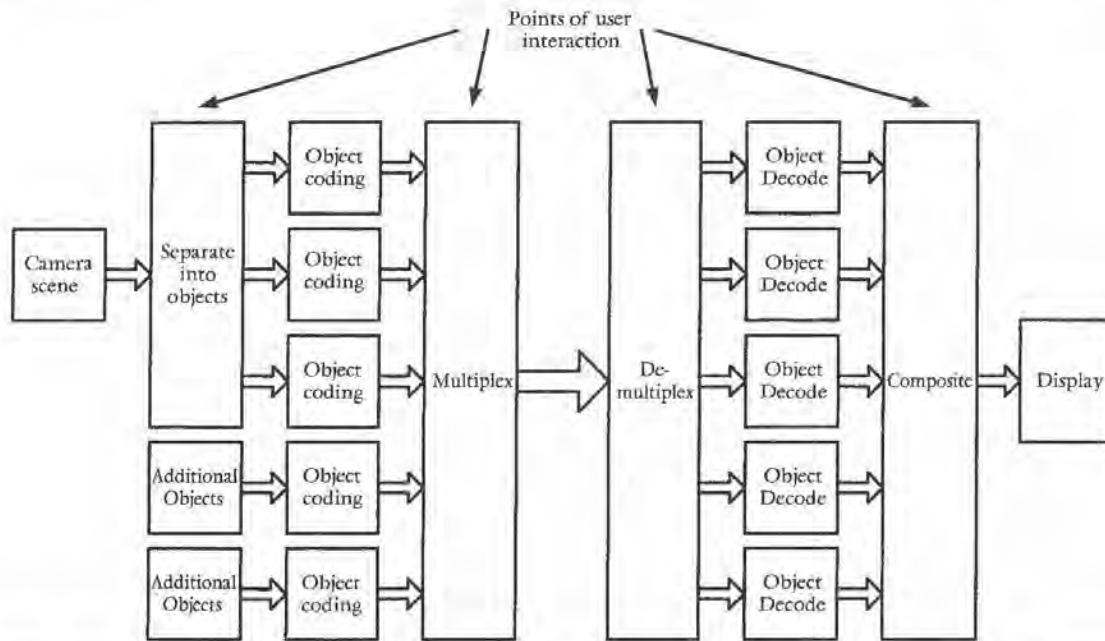


Figure 11.1 Objects and object coding in the MPEG-4 video system.

can be added with a fairly small number of bits. But for this to work, the decoder needs the ability to generate the titles from the information supplied, and to key the titles over the decoded video prior to display. Decoding of MPEG-4 video bitstreams may require a decoder with multiple decode engines and the ability to perform compositing operations.

MPEG-4 is not confined to video and text. We discuss some of the other options later, but for the moment let's continue with the text example. The ability to composite elements at the decoder provides far more than coding efficiency. In conventional systems, the image to be displayed is determined and constructed prior to transmission; overlaid titles become a part of the video prior to encoding and transmission. In MPEG-4, it is possible to transmit several text streams and to choose at the decoder which stream or streams (if any) to combine with the video. The choice may be entirely that of the viewer, or may be constrained by other information in the transmitted bit stream.

A simple and crude example of this capability exists in today's analog television system. "Closed captions" for the hearing impaired are coded into one line at the top of the picture, and the viewer may

choose whether or not to overlay them onto the video. MPEG-4 offers vastly more flexibility and efficiency even with simple text; color, position, and font can all be specified by the author, and the viewer may have the choice, for example, of several languages.

We will look in more depth at the capabilities of the system, but we have already identified three key characteristics of MPEG-4 streams:

- Multiple objects may be encoded using different techniques, and composited at the decoder.
- Objects may be of natural origin, such as scenes from a camera, or synthetic, such as text.
- Instructions in the bitstream, and/or user choice, may enable several different presentations from the same bitstream.

These capabilities do not have to be used—MPEG-4 provides traditional coding of video and audio and improves on MPEG-2 by offering increased efficiency and excellent resilience to errors. However, the true power of MPEG-4 comes from the new applications made possible by the architecture described above. The independent coding of objects offers a number of advantages. Each object may be coded in the most efficient manner, which more closely emulates the processes of the human psychovisual system; the separation of objects allows interaction with these objects—particularly powerful in games and educational programs. Also, different spatial or temporal scaling may be used as appropriate. As an example, if bandwidth and/or computational resources are limited, it may be appropriate to retain full temporal resolution for an important foreground object, but to update to the background at a lower rate.

There are also a number of disadvantages to the object approach. The decoder must be capable of decoding all the possible bit streams it supports, and it must have a compositing capability. Thus a hardware MPEG-4 decoder is necessarily more complex than a comparable MPEG-2 decoder. Similarly, more code is necessary to implement a software decoder, but this decoder can have considerable flexibility in using the limited hardware resources to the best advantage.

Up to now, we have discussed only conventional video images and text. MPEG-4 provides other video objects, some of which are far more innovative; these are discussed in the next section. Clearly a proliferation of object types complicates decoder design, and it would be uneconomical to require all decoders to support all possible constructs and objects. Not surprisingly, MPEG-4 defines a number of profiles and levels, in a similar manner to MPEG-2, permitting implementation with different tool sets and various parameter limitations.

Video in MPEG-4

Before looking at the techniques used for video compression in MPEG-4 we must examine the structure of a video scene, as defined for MPEG-4. A typical scene might include a background, one or more foreground objects such as furniture, one or more people, and some graphic elements. In conventional video thinking, and in MPEG-1 and MPEG-2, the complete scene is sampled once per frame, producing a bitmap that is coded by the techniques discussed in earlier chapters. MPEG-4 can also work this way, but it can also handle each of the objects separately, if the information is presented in the correct way.

To make life simple, let's forget about the furniture for now; we will just assume it is a part of the background. So, apart from any graphic elements, the scene consists of a background and (say) one person as a foreground object. It is well known that the scene may not be that way in the studio—the person, such as a weather forecaster, may be standing in front of a plain blue or green background and *chroma keyed* over a different background (such as a weather map). In the studio, the image of the person in front of the plain-color background is processed to eliminate the background color and to generate a *key signal* or *alpha channel* representing the shape (and, perhaps, the transparency) of the foreground person. This shape information is used to composite the two scene elements together. Where the foreground person stands, the background scene is replaced by the image of the person. Elsewhere (outside the shape) the background is unchanged. If the person is holding a semi-transparent object like a glass of water, the final image must be a mix of foreground and background scenes, determined by transparency information contained in the key signal. For realism, a mix is also needed along the edges of the foreground shapes.

In MPEG-4 terminology, the person in the foreground is known as a *video object*, represented by two elements, the video image of the person, known as the *texture*, and the key signal or alpha channel, known as the *shape*.

The background image does not really need a shape associated with it, but MPEG-4 treats this as having a rectangular shape, a degenerate case.

MPEG-4 Video Hierarchy

The starting point for video in MPEG-4 is a visual concept rather than a signal, such as a foreground person in a scene. Before we can

do anything with the object it must be sampled. Most objects are sampled at regular time intervals (frames), and each time sample (a complete spatial representation at one instant in time) is known as a *video object plane* (VOP). So, each object in the scene is represented by a series of video object planes—except a sprite or other static object, which may be represented by a single video object plane.

In more familiar terms, a camera views a scene and captures the information by sampling the scene (by scanning, or shuttering and scanning). The camera provides its output as a sequence of frames or, in MPEG-4 terminology, the texture part of a sequence of VOPs.

A VOP contains texture data and either rectangular shape information or more complex shape data associated with the object. VOPs, like frames in earlier versions of MPEG, may be coded with intradata, or by using motion compensation; this is discussed below.

Next up is the optional *group of video object planes* (GOV) that groups together video object planes. GOVs are similar to the groups of pictures (GOPs) in earlier MPEGs, and provide points in the bitstream where VOPs are encoded independently from each other, and so provide random access points into the bitstream.

The *video object layer* (VOL) permits scalable coding of a sequence of VOPs or GOVs. Scaling may be spatial or temporal and is discussed below. Multiple VOLs correspond to multiple scalings of a sequence. Each scaling is matched to a set of available resources—usually limited bandwidth and/or limited computational power.

The *video object* (VO) level includes everything in the bitstream about a particular video object. It includes all VOLs associated with the object. Multiple video objects are represented by multiple VOs.

Finally, the *video session* (VS) is the top video level of the MPEG-4 scene, and includes all video objects, natural or synthetic, in the scene.

Shape Coding

There are two types of shape for video objects in MPEG-4, rectangular and arbitrary. *Rectangular shapes* are trivial and merely indicate the extent of an image. Note, however, that this still represents a significant increase in flexibility over earlier standards. In MPEG-2, for example, the extent of the image is hard-coded into the bitstream headers. In MPEG-4, the rectangular size of a simple background video object is comparable, but there can still be other rectangular objects (of differing size) in the same session, such as picture-in-picture, and sprites.

Arbitrary shapes are the MPEG-4 equivalent of the graphics alpha channel. The shape again represents the extent of a video object, and at any point in the image plane it determines whether its associated object may be visible. The shape is defined over a rectangular area, known as the *mask*, sized to accommodate the greatest horizontal and vertical dimensions of the object itself. Both the horizontal and vertical sizes of the shape mask are multiples of 16 pixels.

Arbitrary shapes may be coded as binary data or grayscale data. Binary shapes are the simplest and indicate whether the object is transparent or opaque at any given point. The edge between the object and background is always hard. This normally represents a violation of Nyquist, and aliasing is likely to be visible. Grayscale shapes, on the other hand, permit a smooth, bandwidth-controlled transition or "blend" between objects and can yield much greater realism.

Binary shapes are used for simple applications and can represent total transparency or total opacity. In image areas where the video object is determined to be transparent, it cannot be seen under any circumstances. The background, or the composite formed by the background and any lower-level video objects, will be seen (unless occluded by some other higher-level object). Where the video object is opaque, the background and any lower-level objects will be invisible, and the video object associated with the shape will be visible, unless occluded by a higher-level object (something in front).

Binary shapes are coded into blocks 16 pixels square, like the macroblock used for textures, but they are known as *binary alpha blocks* (BABs). There are three classes of block in a binary mask; those where all pixels are transparent (not part of the video object); those where all pixels are opaque (part of the video object); and those where some pixels are transparent and others opaque. Coding of the first two cases is trivial. Blocks of the third type are those representing the boundary of the video object and are coded using techniques derived from arithmetic coding, discussed briefly in Chapter 3.

The algorithm used is known as *context-based arithmetic encoding* (CAE), extended to include the use of motion estimation. As in the conventional arithmetic encoding algorithm, coding is based upon a continually updated probability estimate. In the basic interCAE, the probability estimate is computed from ten pixels, above and to the left of the pixel being coded, see Figure 11-2a. For intraCAE, where motion estimation and prediction are used, the context includes some pixels from the current frame and some from the reference frame, as seen in Figure 11-2b.

Figure 11-2a
Context for intraCAE.

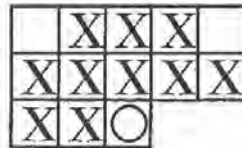
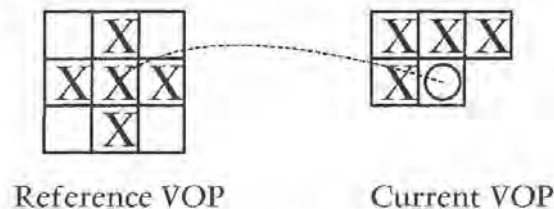


Figure 11-2b
Context for interCAE.



Grayscale shapes are represented much as a luminance signal is, usually quantized to eight bits, with values from 0 (totally transparent) to 255 (totally opaque). The shape is coded using a philosophy similar to that used for binary shapes. Again there are macroblocks that are totally outside the boundary of the object, and those that are totally within the boundary of the object. Blocks outside are marked as "all zero" (fully transparent); those inside are marked as "all 255" (fully opaque). Blocks that are neither fully transparent nor fully opaque must be coded in a manner similar to that used for texture; that is, motion compensated DCT.

Texture Coding

Texture coding, the MPEG-4 term that corresponds to coding of conventional moving-image data, is built on MPEG-2 coding with extensions and improvements, as MPEG-2 was built on MPEG-1 and JPEG. In fact, a rectangular VOP is the closest equivalent to a "video frame" in pre-MPEG-4 terminology. The same concepts of intra- and intercoding apply, and video objects may be coded with I-, P-, and B-VOPs. All MPEG-4 profiles, except for the studio profiles described below, use a 4:2:0, YUV representation of video object textures.

In MPEG-4, not all video objects are the same size, and texture coding is necessary only for those areas that are part of the object. For rectangular objects this is simple; the size must be a multiple of 16 pixels

(one macroblock) in each direction, and all macroblocks are processed. For objects with a more complex shape, the boundary is defined by the shape signal. The extent of the object is still defined by a rectangular array of macroblocks, but texture coding is performed only for those macroblocks wholly or partially inside the object boundary.

I-VOPs are coded much as MPEG-2 I-frames are, but several new techniques have been introduced to improve efficiency. In Chapter 4, we looked at predictive coding, and in Chapter 10, we saw that MPEG-2 used the simple $\hat{X} = A$ predictor for encoding DC coefficients and motion vectors. Chapter 4 also showed that there are much better predictors, and that the performance of a simple adaptive predictor was substantially higher than that of other one- or two-dimensional predictors. MPEG-4 uses an adaptive predictor for the DC values. The predictor measures horizontal and vertical brightness gradients and predicts the DC value from the block above or the block to the left, in the direction of the lesser gradient.

Just as the correlation of the image provides benefit by predicting DC coefficients, it can also help in the coding of some AC coefficients. Areas of similar texture (in the conventional sense of the word) generate similar arrays of AC coefficients after the DCT transform. The similarity (and, therefore, the coding benefit) is greatest in the most significant coefficients; those that represent the greatest energy of the texture. These coefficients are normally the non-zero coefficients in the first row and/or first column. These are also the coefficients that will be quantized least aggressively and use up the most bits for transmission and thus offer the greatest potential for improved coding.

In MPEG-4, the AC coefficients of either the first row or the first column are predicted from those of the block immediately above, or immediately to the left, or above-left.

Quantization of the coefficients is similar to the method used by MPEG-2, but the mechanisms for coefficients scanning, and for variable-length encoding, are both improved.

The chosen method for coefficient readout is determined by the DC prediction. When there is no DC prediction, the zigzag scanning, as described under MPEG-2, is used. If the DC coefficient was predicted from the block to the left, *alternate-vertical* scan is used, a scanning system that emphasizes reading out the vertical direction first. If, however, the DC coefficient was predicted from the block above, *alternate-horizontal* scan is selected, which results in a readout of coefficients with an emphasis on the horizontal direction first.

To improve the efficiency of variable-length encoding, two different VLC tables are provided, and the quantization level determines which table is used. The VLC codes themselves are reversible. This is part of the error-resilience strategy of MPEG-4; if there is an error in the received bit stream, decoding can continue up to the error. Data after the error may be decoded by starting at the end of the block and decoding the VLC codes in reverse until the error is reached.

Boundary Coding

The ability to code objects of arbitrary shape creates an interesting situation at the object boundaries. Blocks that lie outside the boundary of the object need no texture coding. Those that are totally inside the boundary are coded normally according to the techniques we have discussed. Texture coding is needed for boundary blocks, but the object only exists for part of the block. If we chose to make the pixels outside the boundary of the object black, this would introduce a good deal of coefficient energy when the block was DCT transformed, yet this energy would not represent a meaningful part of the image. To avoid this inefficiency boundary blocks are *padded*. First, all pixels that are not part of the object are given a value equal to the average value of all the pixels that are part of the object. The padding is refined by stepping through the pixels outside the object and performing a correction based on the average value of any neighbors that lie within the object. Note that no values within the object are changed. Changes to values outside the object do not affect the final result because these pixels will never be visible. The process described minimizes the energy of the coefficients when the block is DCT transformed.

Coding of Arbitrary-Shaped Video Objects

Now that we have looked at coding the different elements, it is worthwhile to examine all the possible cases for block coding within the mask of a video object.

In the simplest possible case, the block is flagged as transparent in the shape part of the VOP. This means that all the pixels in the block lie outside the boundary of the object, so that no texture coding is required.

The block may be flagged as opaque in the shape part of the VOP. In this case, all pixels in the block lie inside the boundary of the object. No boundary coding is required, but texture coding is needed. The texture may be intracoded or predicted with motion compensation, as appropriate.

The remaining blocks are in some way part of the boundary area of the object. In the case of a binary shape this means there are some pixels that are transparent and some that are opaque. In the case of a grayscale shape, it means that not all pixels are transparent and not all pixels are opaque. Here both shape and texture must be coded. Coding of both the shape and texture may be intra or predictive.

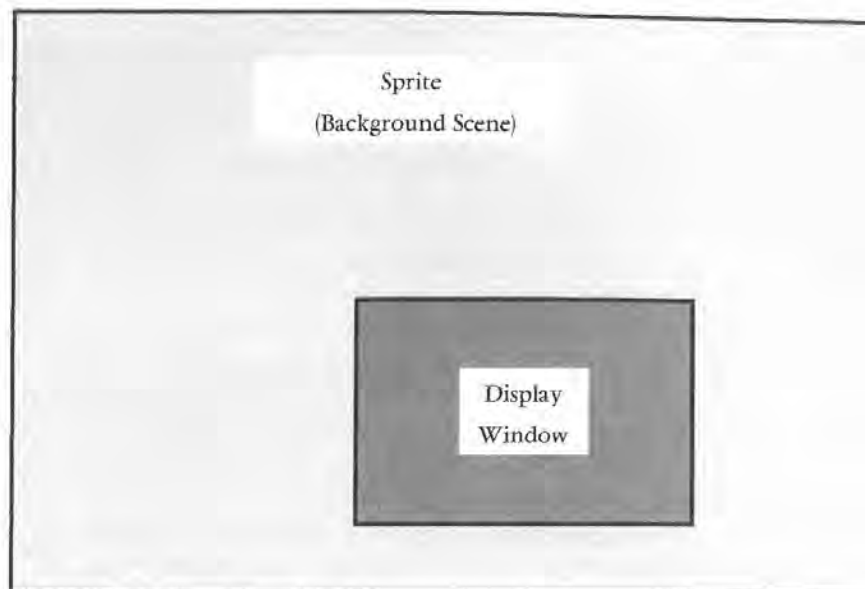
Just as in earlier systems, the syntax must support many coding possibilities. Any block may be intracoded; in a P-VOP or B-VOP, blocks may be predictively coded. In the simplest case, the *motion vector difference* (MVD) is zero (i.e., the motion-vector prediction is perfect), and the motion vector points to a good match for the block, so that no residuals are required. This block just requires a "skip" code. If the match is not sufficiently good, residuals must be transmitted. In other cases, the MVD will be nonzero and residuals may or may not be required.

Sprites

MPEG-4 has another object type that is particularly useful for backgrounds; the sprite. A *sprite* is a video object, usually larger (perhaps considerably larger) than can be displayed at any one time. It is used for an object (like a static background) that is persistent through a scene. Although sprites are not restricted to games, this application provides their most obvious use. Generally a game scene consists of a background plus a number of (usually) synthetic objects that move according to the script of the game and the actions of the player. During the action, the viewed scene will likely pan around the background; the player sees different parts at different times, but they are all parts of the same static image. MPEG-4 provides the ability to transmit the whole of the background once as a sprite, and to generate scenes by sending cropping or warping information to determine which part of the sprite is visible at any given instant (Figure 11-3). Once the sprite has been transmitted, only the cropping/warping information for the sprite and the foreground objects need be transmitted. In a typical game, each section of the sprite will likely be used many times, so this approach represents a substantial reduction in the total data used.

Figure 11-3

The sprite provides a background scene larger than the display window; once all the sprite has been transmitted, any part of it may be displayed.



Transmitting the whole of a sprite at the beginning may be very efficient, but will need either extra bandwidth or a period of time before the action can start. MPEG-4 has techniques to avoid this problem as well. A sprite can be transmitted in sections, as needed. The opening scene would require that only the area of the sprite needed for that scene be transmitted immediately. If the "camera" then panned to the left, a new strip of information would be transmitted every frame, but these would all be stored at the decoder as part of the sprite. So, if the pan reversed, no new background information would need to be transmitted until the camera moved to the right of the original scene. However, a certain amount of sprite data could be transmitted with each frame anyway, as bandwidth permitted, to continue building the complete image at the decoder. This method still requires that a full screen of data be transmitted before the first scene may be viewed. If this is unacceptable, the sprite may be *progressively* encoded, so that only a low-resolution version is transmitted at first; more area and more resolution are sent later.

Sprites are encoded as luminance plus two color components, as we have seen in JPEG and earlier versions of MPEG, and are always intra-coded, since the image is essentially static. However, I-encoding uses the improved techniques introduced in MPEG-4, discussed in the previous section.

Static Texture Coding

MPEG-4 includes the ability to map a static texture onto a varying shape. For example, a logo or message may be mapped onto a swimming fish. This operation is performed at the decoder; the moving shape and the static texture are transmitted as separate objects. In this case the ability to scale the static texture is very important. The movement of the shape may cause stretching or expansion of the texture far beyond its nominal size. To maximize the quality of this operation, MPEG-4 provides a wavelet-based tool for compression of the static texture. Wavelets are discussed in a later chapter; the important point here is that wavelets provide good scaling capability; objects that are expanded display minimal artifacting.

Animations

As discussed above, one of the great strengths of MPEG-4 is the ability to transmit both natural and synthetic objects that may be composited at the decoder. One of the more interesting capabilities using synthetic objects is *facial animation*. This is another example of mapping a texture onto a moving shape, but in this case the shape is specified by a mesh or 3-D model specified by a number of *nodes*. The position of each node is coded, and again predictive coding may be used to increase the efficiency of the coding as the face-shape changes. The standard includes all the syntax necessary to code the position and movement of the nodes. It also includes higher-level constructs, such as *visemes*, or visual lip configurations equivalent to speech phonemes.

With sufficiently good implementation, this technique opens up some fascinating (or frightening) possibilities. A real or imaginary person may be represented by a mesh and a texture. In combination with speech synthesis, the animated face can be coded to "read" any textual message.

Version 2 of MPEG-4 goes on to add body animation capability. The following extract from the official Overview of the MPEG-4 Standard is reproduced by kind permission of the MPEG Convener. Not only does this short quote give a description of the body-animation capability, but I think it provides an excellent example of the thoroughness of every aspect of MPEG-4.

The Body is an object capable of producing virtual body models and animations in the form of a set of 3-D polygonal meshes ready for rendering. Two sets of parameters are defined for the body: body definition parameter (BDP) set, and body animation parameter (BAP) set. The BDP set defines the set of parameters to transform the default body to a customized body with its body surface, body dimensions, and (optionally) texture. The body animation parameters (BAPs), if correctly interpreted, will produce reasonably similar high level results in terms of body posture and animation on different body models, without the need to initialize or calibrate the model.

Upon construction, the Body object contains a generic virtual human body with the default posture. This body can already be rendered. It is also immediately capable of receiving the BAPs from the bitstream, which will produce animation of the body. If BDPs are received, they are used to transform the generic body into a particular body determined by the parameters contents. Any component can be null. A null component is replaced by the corresponding default component when the body is rendered. The default posture is defined by standing posture. This posture is defined as follows: the feet should point to the front direction, the two arms should be placed on the side of the body with the palm of the hands facing inward. This posture also implies that all BAPs have default values.

No assumption is made and no limitation is imposed on the range of motion of joints. In other words the human body model should be capable of supporting various applications, from realistic simulation of human motions to network games using simple human-like models. The work on Body Animation includes the assessment of the emerging standard as applied to hand signing for the listening-impaired.

Scalability

MPEG-4 provides both spatial and temporal scalability at the object level. In both cases, this technique is used to generate a *base layer*, representing the lowest quality to be supported by the bitstream, and one

or more *enhancement layers*. These layers may all be produced in a single coding operation. The scaling can be implemented in two different ways. When there are known bandwidth limitations, versions of the bitstream may be used that include only the base layer, or the base layer plus lower-order enhancement layers. Alternatively, all layers may be sent and the scaling decision left to the decoder. If the display device is low resolution, or if computational resources are insufficient, enhancement layers may be ignored.

Figure 11-4 shows the concept of an encoder implementing spatial scalability; in this case at just two levels. The input VOP is down-converted to a lower resolution, the base layer. This layer is encoded, and then a decoder constructs the base-layer VOP as it will appear at the decoder. This VOP is then up-converted to the same resolution as the input, and a subtraction process yields the differences from the original image, these are separately encoded in an enhancement-layer encoder. Note that each stream of encoded VOPs forms a video object layer. The base-layer VOL uses both intra- and intercoding, but the enhancement layer uses only predictive coding. The base-layer VOPs are used as references, as shown in Figure 11-5.

Figure 11-4
Spatially scalable
encoder for a single
enhancement layer.

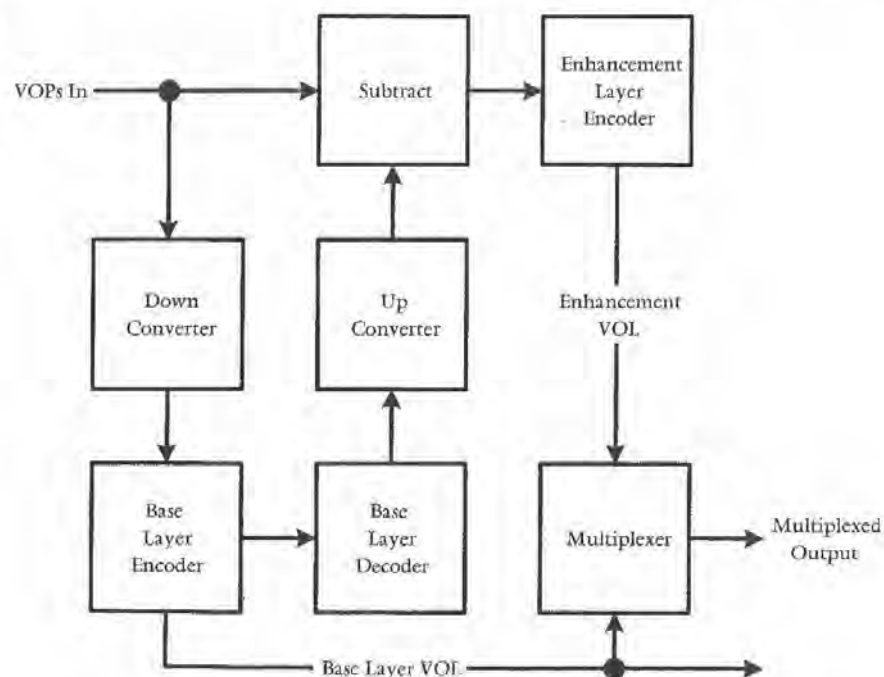
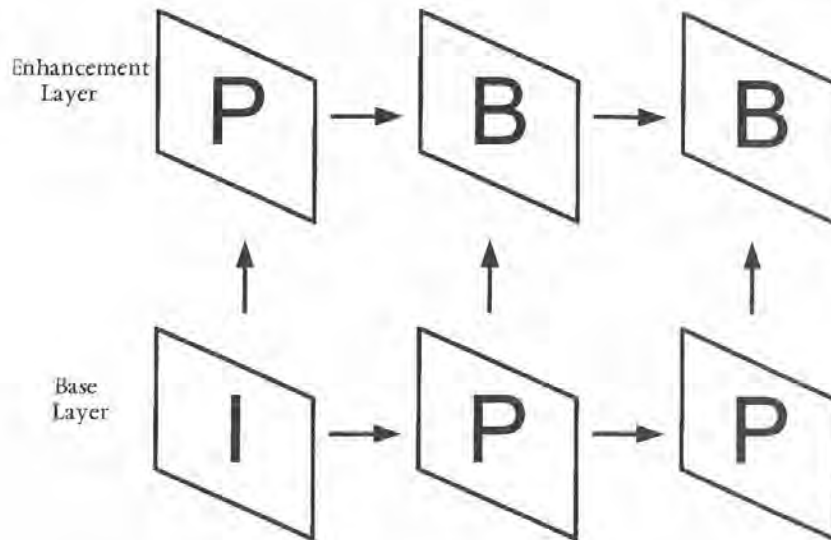


Figure 11-5
Prediction of a spatial
enhancement layer.



Temporal scalability is rather more simple. The stream of incoming VOPs is split. The required number of VOPs (perhaps every third VOP) is sent to the base-layer encoder; the remainder is sent to one or more enhancement layers.

Note that scalability is applied on a per-object basis. This provides a wide range of coding and decoding capabilities. For example, a decoder in a game system might have insufficient computational power to decode all objects at the highest available rate. It might choose to decode only a low rate for the background, accepting some level of jerky motion there, but to decode a higher rate for the principal foreground object, making its motion as smooth as possible.

Advanced Coding Extensions (ACE)

Version 2 of MPEG-4 introduced three new tools to improve coding efficiency for video objects. These are *global motion compensation* (GMC), *quarter pel (pixel) motion compensation*, and *shape-adaptive DCT*. Together, these tools have been shown to improve coding efficiency by up to 50 percent over Version 1, depending on the type of material and the bit rate.

Global motion compensation and quarter pixel motion compensation are fairly self-explanatory. GMC allows the overall motion of the

object to be coded with a very few parameters, and the improved resolution of motion vectors substantially reduces prediction errors and the use of residuals.

Shape-adaptive DCT may be used to improve the coding efficiency of boundary blocks, when not all pixels belong to the image. Instead of using 8×8 two-dimensional DCT on the complete block of 64 pixels, one-dimensional DCT is applied first vertically, then horizontally, but only to the pixels that belong to the object; these are called the *active pixels*.

To start the process, each column of the block is examined, and within each column all the pixels that belong to the object are moved up, so the column is "top-justified." Then, each column is DCT-transformed, but using a DCT matched in size to the number of pixels. For example, if there are five pixels in a column that belong to the object, a 1×5 DCT would be used for that column. If there are no active pixels in the column, no transform is performed. When the vertical transforms are complete, the active pixels are shifted to the left, so that each row is "left aligned." A one-dimensional DCT is then performed on each row that has active pixels, again using the appropriate size of transform.

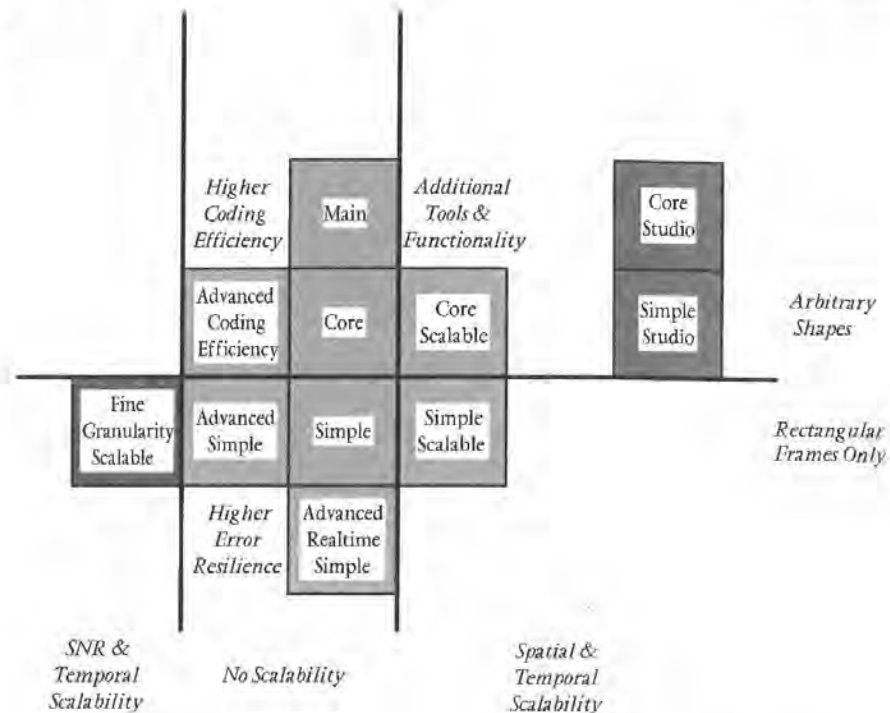
Because the contour of the object is transmitted separately by shape coding, the pixels can be moved back to the "the right place" in the decoder.

Visual Profiles

MPEG-4 includes a structure of profiles and levels using the same philosophy as MPEG-2. As with all aspects of MPEG-4 the range is vast. Rather than paraphrase, I include part of the listing from the Official Overview; this extract covers visual profiles only for Versions 1 and 2. Figure 11-6 shows the relationships among the various profiles and includes the two studio profiles defined by the proposed Amendment 3, and the Fine Grain Scalability profile that is the subject of proposed Amendment 4. These additions to the standard are discussed below. Figure 11-6 is reproduced from the work of Professor Jens-Rainer Ohm of the University of Aachen, with his kind permission.

The visual part of the standard provides profiles for the coding of natural, synthetic, and synthetic/natural hybrid visual content. There are five profiles for natural video content:

Figure 11.6
Profiles in MPEG-4.



1. The *Simple Visual Profile* provides efficient, error-resilient coding of rectangular video objects, suitable for applications on mobile networks, such as PCS and IMT2000.
2. The *Simple Scalable Visual Profile* adds support for coding of temporal and spatial scalable objects to the Simple Visual Profile. It is useful for applications that provide services at more than one level of quality due to bit-rate or decoder resource limitations, such as Internet use and software decoding.
3. The *Core Visual Profile* adds support for coding of arbitrary-shaped and temporally scalable objects to the Simple Visual Profile. It is useful for applications such as those providing relatively simple content interactivity (Internet multimedia applications).
4. The *Main Visual Profile* adds support for coding of interlaced, semitransparent, and sprite objects to the

Core Visual Profile. It is useful for interactive and entertainment-quality broadcast and DVD applications.

5. The *N-Bit Visual Profile* adds support for coding video objects having pixel depths ranging from 4 to 12 bits to the Core Visual Profile. It is suitable for use in surveillance applications.

The profiles for synthetic and synthetic/natural hybrid visual content are:

1. The *Simple Facial Animation Visual Profile* provides a simple means to animate a face model, suitable for applications such as audio/video presentation for the hearing impaired.
2. The *Scalable Texture Visual Profile* provides spatial scalable coding of still image (texture) objects useful for applications needing multiple scalability levels, such as mapping texture onto objects in games, and high-resolution digital still cameras.
3. The *Basic Animated 2-D Texture Visual Profile* provides spatial scalability, SNR scalability, and mesh-based animation for still image (textures) objects and also simple face object animation.
4. The *Hybrid Visual Profile* combines the ability to decode arbitrary-shaped and temporally scalable natural video objects (as in the Core Visual Profile) with the ability to decode several synthetic and hybrid objects, including simple face and animated still image objects. It is suitable for various content-rich multimedia applications.

Version 2 adds the following Profiles for natural video:

1. The *Advanced Real-Time Simple (ARTS) Profile* provides advanced error resilient coding techniques of rectangular video objects using a back channel and improved temporal resolution stability with the low buffering delay. It is suitable for real time coding applications; such as the videophone, teleconferencing, and remote observation.

2. The *Core Scalable Profile* adds support for coding of temporal and spatial scalable arbitrarily shaped objects to the Core Profile. The main functionality of this profile is object-based SNR and spatial/temporal scalability for regions or objects of interest. It is useful for applications such as the Internet, mobile, and broadcast.
3. The *Advanced Coding Efficiency (ACE) Profile* improves the coding efficiency for both rectangular and arbitrary shaped objects. It is suitable for applications such as mobile broadcast reception, the acquisition of image sequences (camcorders), and other applications where high coding efficiency is requested and small footprint is not the prime concern.

The Version 2 profiles for synthetic and synthetic/natural hybrid visual content are:

1. The *Advanced Scalable Texture Profile* supports decoding of arbitrary-shaped texture and still images including scalable shape coding, wavelet tiling, and error-resilience. It is useful for applications that require fast random access as well as multiple scalability levels and arbitrary-shaped coding of still objects. Examples are fast content-based still image browsing on the Internet, multimedia-enabled PDAs, and Internet-ready high-resolution digital still cameras.
2. The *Advanced Core Profile* combines the ability to decode arbitrary-shaped video objects (as in the Core Visual Profile) with the ability to decode arbitrary-shaped scalable still image objects (as in the Advanced Scalable Texture Profile). It is suitable for various content-rich multimedia applications such as interactive multimedia streaming over Internet.
3. The *Simple Face and Body Animation Profile* is a superset of the Simple Face Animation Profile, adding—obviously—body animation.

Scene Compositing and Interaction

Scene Modeling

Although it is not directly related to video compression, it would be wrong to discuss MPEG-4 without some mention of *Binary Format for Scenes* (BIFS). The introduction to this chapter highlighted the fact that MPEG-4 can handle multiple video objects, all transmitted independently, but composited at the decoder. The same philosophy is applied to audio. A complete MPEG-4 scene may include multiple audio objects, natural and synthetic, such as background sound, dialogue, sound effects, etc. There can also be multiple 2-D and 3-D graphical objects in a scene.

BIFS is the language that describes, statically and dynamically, how objects should be brought together at the decoder to form a complete scene. It is based on *virtual reality modeling language* (VRML) with extensions that provide simpler constructs for 2-D objects and coordinate spaces. BIFS provides a hierarchical or tree structure wherein objects may be combined into groups. Individual objects may be manipulated within a group, or the whole group may be manipulated. Nodes of the tree may be added or removed at any time.

Each object and each group have a local coordinate system, addressing both space and time parameters. All of these coordinate systems relate to a set of global coordinates describing the scene, so that any object may be manipulated with respect to itself, or within any layer of a group structure, or with respect to the overall scene. Thus, BIFS provides a timeline within which any part of the scene can be choreographed. BIFS data is sent in a separately identified stream or multiplexed with video or audio sessions.

MPEG-4 Version 2 includes some extensions to BIFS, known as *advanced BIFS*. To demonstrate the richness of MPEG-4, I again quote a paragraph from the Official Overview. To stress that the visual schemes discussed above are just one part of MPEG-4, this extract refers to Advanced BIFS constructs for the audio parts of a scene:

Advanced sound environment modeling in interactive virtual scenes, where properties such as room reflections, reverberation, Doppler effect, and sound obstruction caused by objects appearing between the source and the listener are computed for sound sources in a dynamic environment in real time. Also enhanced source directivi-

ty modeling is made possible enabling inclusion of realistic sound sources into 3-D scenes.

Interaction

MPEG-4 is designed for use in applications where the end-user may interact with the scene presented. The most obvious application example is games, but interaction is a powerful tool that is an essential element of any system where more information is available than can be presented at one time. Interactive television, educational systems, any sophisticated data retrieval system; all need the ability to change the presentation based upon user requests—just as on a Web site.

Interaction may be provided using many devices—mice, remote controls, and keyboards are just a few. Fortunately, the MPEG-4 constructs do not require additional standardization or a guess at future mechanisms for interaction. Any interactive device is required to produce BIFS commands, and these are combined with the BIFS content of the master scene. The decoder need not know whether the commands were part of the original coding or contributed by an interaction mechanism; the same capabilities exist.

Work in Progress

During September 2000, the MPEG committee is completing work on two more amendments to MPEG-4, scheduled to become part of the standard during 2001. As with earlier work, the new amendments are true supersets, extending the application of the Standard into new areas.

Studio Profiles

I mentioned at the beginning of this chapter that MPEG-4 had expanded far beyond its original focus of low-bit-rate applications. *Studio profiles* are intended for use in production environments and for the storage and transport of program material up to the point of transmission to the end-user, where an appropriate profile and level would be chosen for the application. It is important to note that the

MPEG-4 studio profiles are true members of the MPEG-4 system, not just extensions of MPEG-2. In fact, the object-oriented structure is seen as a natural fit with the production process. If, for example, video objects can be kept separate, and their compositing is described separately, there are no irrevocable steps. The original material may be re-edited or repurposed without limitation.

The MPEG-4 studio profiles overlap MPEG-2, in that they include the capabilities of the MPEG-2 4:2:2 profile. MPEG-4, however, goes much further, and the studio profiles include most of the powerful tools and constructs specified for the other profiles, together with extension into higher-definition formats and other areas critical to high-quality production.

The studio profiles add a few more tools to improve efficiency in the high-quality domain. In particular, DCT coefficients may be grouped by value, and variable-length coding uses recursive selection of the most efficient of a number of VLC coding tables. These tools offer an increase in coding efficiency of around 10 percent when compared to MPEG-2.

Two studio profiles are specified. Both provide for 4:2:2 or 4:4:4 coding in either the *YUV* or *RGB* domains. They provide for picture sizes of up to 4k×2k (4000×2000 pixels), progressive or interlace, and pixel depths of up to 12 bits per component. Encoded data rates can be up to 1.2 Gbits/s. Both studio profiles support arbitrarily shaped video objects.

The *Simple Studio Profile* is intended for acquisition and editing environments. The profile permits only intracoding, so that frames or VOPs are always independent, and the video can be accessed at any point by an editor without complication or loss of temporal granularity. The Simple Studio Profile also permits lossless transcoding from the 4:2:2 profile of MPEG-2.

The *Core Studio Profile* is intended for distribution and storage within production environments. It includes all of the elements of the Simple Studio Profile, plus sprites, and permits predictive encoding, using P-VOPs only. There is no bidirectional prediction within MPEG-4 studio profiles.

Fine Grain Scalability

The *Fine Grain Scalability Profile* (FGS), often known as the *Streaming Profile* is intended to support applications and environments where

the bandwidth and/or computational power available cannot be predicted and may vary dynamically. The scaling tools provided by other parts of MPEG-4 are powerful in their own way, but have two significant limitations.

The granularity of scaling using existing tools is necessarily coarse. Earlier, we saw a process that could produce a 2-layer bitstream including a base layer and one enhancement layer. The number of enhancement layers that can be used is arbitrary, but in practical terms is severely limited. Each layer adds coder and decoder complexity and overhead in the bitstream. Also, provision for all the levels of scalability must be built in at the time of encoding; any changes require recoding from scratch.

In contrast, FGS provides a mechanism that permits a single encoding process, producing a bitstream that can be modified subsequently in two different ways. Prior to transmission, the bitstream may be processed to scale it down to known bandwidth limitations. This can be performed dynamically, for example in response to the requirements of a statistical multiplexing system. Downstream distribution points may further reduce the bit rate if necessary. At the receiving end, the decoder can use all of the received bits, or it can scale the bitstream down further to match its capabilities and resources, again dynamically.

Figure 11-7 shows the concept of bit-plane encoding. The quantized DCT coefficients are indicated by the columns of bits. Clearly the most significant coefficient is represented by the tallest column. We can "slice" the bit planes horizontally. If we took only the highest plane in the figure, we would capture only the MSB of the largest coefficient. Our representation of the block would then be a single coefficient with the quantized value 1000 (instead of 1110). This is obviously a very crude representation, but it is better than nothing.² If we were able to take two bit planes, we would get three coefficients, one with the value 1100, and two each with the value 100. The important point is that from a single coding operation we can extract at the decoder as much information as we are able to use from the bitstream, and be assured that we have captured the most significant elements. This approach is also known as *signal-to-noise scalability*.

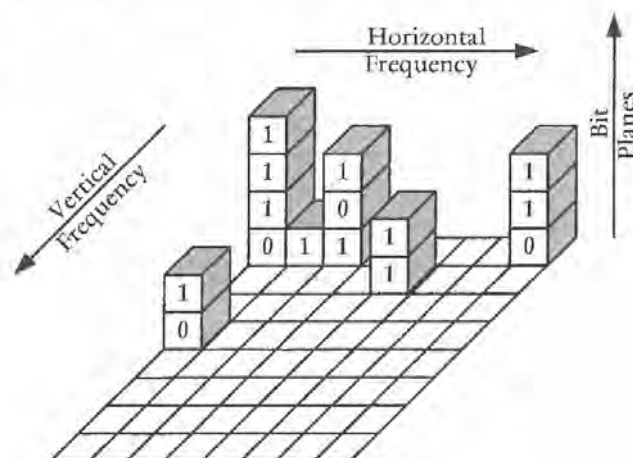
It is possible to modify this technique at the encoder. Some values may be artificially pushed up the bit-plane stack. For example, it is possible to apply frequency weighting to the DCT coefficients,

² As discussed in Chapter 6, the reconstruction value would likely be different. For an effective range of 1000 to 1111, the probable reconstruction value is 1100.

emphasizing the importance of the low-frequency coefficients, or to increase artificially the value of all coefficients that lie within a particular *region of interest* in the object.

The streaming profile is, of course, especially useful in Internet environments. Fine-grain scalability may be used alone, or combined with temporal scaling. A session may be multicast by a server at the highest quality permitted by the bandwidth available. Downstream servers may, if necessary, trim the bitstream to suit lower-bandwidth parts of the network. A decoder uses only as much of the bitstream as its resources permit. It may ignore temporal enhancement layers to minimize the number of VOPs to be processed, and it can take only as much data as it can handle from the other layers.

Figure 11-7
Bit planes in an array
of DCT coefficients.



Conclusion

I think this is the longest chapter in the book, and those who are familiar with MPEG-4 will know that I have only scratched the surface of this Standard and its various amendments. I have tried to cover, at least briefly, those elements of the standard most relevant to video compression, and to convey an impression of the vast scope of MPEG-4. For those interested in learning more, MPEG has become much more open to accessibility of documents (prior to the final Standard, which is controlled by ISO), and its Web site carries many working documents and tutorials.

CHAPTER **20**

**Audio
Compression**

Introduction

This will be a short chapter, partly because of limited knowledge on my part, but also because audio compression is built on many of the same principles as video compression. Only in one sense is audio compression radically different from video compression—it is much older and has been commercially exploited in the analog domain. It was not called *compression*; it was known as *noise reduction*, but essentially these are the same. A noise reducer modifies the signal such that when reverse processing is applied after transmission or recording, the noise of the transmission or recording channel is attenuated. The well known Dolby A, B, and C systems are actually simple, analog, sub-band companding systems—terms we encounter below.

Nevertheless, our interest is on compressing digital audio. Depending on the quality requirements, audio may be sampled at a variety of frequencies, usually in the range 8 to 48 kHz, and with sample resolutions from 8 to 24 bits. The lower limits provide telephone-grade audio; the higher can match or exceed human hearing capabilities. Of most interest in the professional world are two standards at the high end. Compact disc audio is sampled at 44.1 kHz, with a word length of 16 bits for each of two channels, a total of just over 1.4 Mb/s. Professional studio audio is usually sampled at 48 kHz with a word length of 20 bits, almost 1 Mb/s per channel. This rate can accommodate audio frequencies to over 20 kHz with a dynamic range of 120 dB.

As with any form of compression, the objective is to reduce the data by eliminating redundancy. There is “absolute” redundancy—data can be removed by an encoder then unambiguously recreated by a decoder. Removal of this redundancy is lossless compression. There is also perceptual redundancy—data can be removed without significant change to the experience of a human observer.

Perceptual redundancy itself falls into two categories. There are phenomena to which the observer is intrinsically insensitive. In video, we saw that the higher-frequency transform coefficients could be coarsely quantized without significant impact on the viewer. This is true for all conditions. Another effect, used in bit allocation in video systems, is masking. An artifact that might be visible and objectionable in an image area with little activity (empty sky, for example), becomes invisible or insignificant in a busy area of an image.

All these effects exist in audio, and all are exploited by compression systems, but masking is by far the most significant contributor to data

savings in a single-channel audio system. (Multiple channel systems also benefit from correlation between the various channels.) To understand the mechanisms of masking in the human hearing process we must examine the workings of the inner ear.

Masking in Human Hearing

The human hearing system has remarkable performance. Its useful range covers about 10 octaves and, in the more sensitive regions, it provides a dynamic range of more than 100 dB. The discriminating power of the ear—brain combination is incredible; we can resolve conversation and other intelligence from noise levels and interfering signals to a degree that seems to defy the laws of physics. This performance requires some very sophisticated processing.

One of the key elements of the ear is the basilar membrane of the inner ear. It separates two of the fluid-filled chambers of the cochlea and is in contact with the hairs of the organ of Corti. These hairs drive the actual sensors that send nerve messages to the brain. The basilar membrane varies in width, thickness, and rigidity along its length. This makes it frequency sensitive; different areas vibrate at different frequencies. The basilar is not just a passive membrane; it has an active mechanism that can provide positive feedback to low-amplitude vibrations. The combination of these characteristics leads to interesting behavior. (This explanation is somewhat simplistic. Various authorities have suggested different mechanisms for the behavior of the inner ear, particularly in respect to the active attributes. Some suggest nonlinearity in the cochlea; others attribute the active element to the organ of Corti. Suffice it to say that the behavior is complex, certainly not linear, and that there are active elements within the inner ear. We concentrate on the effects of this complexity.)

Some authorities estimate that there are 24 regions in the basilar membrane; others suggest a higher number, but it is certainly finite. Each region can vibrate over a small range of frequencies, but at only one frequency at a time within that range. When positive feedback is applied, the effect is that of a very high-Q tuned circuit. Each region vibrates at a frequency determined by the strongest stimulus within its range, *and is unaffected by any smaller stimuli*. This means that within each frequency band, only the loudest frequency contributes to the signal received by the brain, and thus what we hear.

This effect is the single most significant contributor to our ability to compress audio.

The effect is called *frequency masking*, and it contributes to our ability to compress in two distinct ways. As mentioned above, only the strongest stimulus within a region is significant. Any frequency in the signal in the same band, but lower in amplitude, need not be coded. Also, noise within the band is irrelevant, provided it is sufficiently below the prime stimulus. This allows encoding of the prime stimulus with a relatively small number of bits; the resulting quantization noise is masked just like any other noise. This effect is quite remarkable and, thanks to Dolby Laboratories, there is a demonstration on the CD-ROM that is well worth trying.

The positive feedback of the basilar contributes to another effect, again similar to that of a high-Q tuned circuit. The vibration responds slowly to changes in the amplitude of the stimulus, and this results in *temporal masking*—we fail to hear sounds a short time before, and a longer time after, a strong stimulus.

Simple Audio Compression Schemes

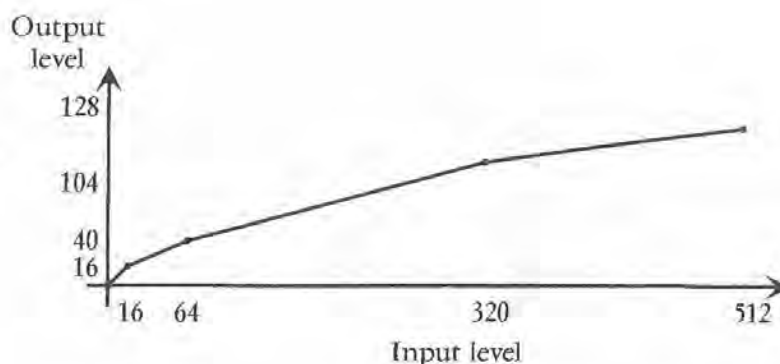
Many of today's consumer devices use some form of audio compression. Sometimes this is used to increase storage capacity. One example is found in 8-mm video recorders; compression is employed to reduce the tape area needed for audio. Audio workstations become much more cost effective if the capacity of the hard drives can be increased by using compression, provided the necessary high quality can be maintained.

In some applications, the compression is used for more than one purpose. The mini compact disc achieves its recording density by means of compression, but data reduction is also used to improve replay under adverse conditions. Data is read from the mini CD at the same speed as from a regular CD—about twice as fast as necessary for normal replay. The data enters a buffer, and after the buffer delay (typically a few seconds) it is decompressed and presented to the listener. If the listener is an energetic jogger and jars the player enough to move the read head, the player can continue to play from the buffer while the head is repositioned, and the additional data speed allows the buffer to be filled again, ready for the next "jog."

Simple schemes such as these generally use some form of *companding*. The number of bits required by each word is reduced by dynamically altering the level or gain according to the instantaneous amplitude, or the amplitude range of a group of samples.

Video 8, for example, uses a simple four-step gain control based on instantaneous signal amplitude. Low amplitudes are passed at unity gain; higher amplitudes are gain-reduced by factors of 2, 4, or 8. The result looks to a video engineer like a gamma curve (see Figure 20-1). The higher levels are coarsely quantized, but the increased quantization noise is masked by the high signal level. (In the case of Video 8, this mild reduction in data rate is supplemented by an analog compressor.)

Figure 20-1
Transfer function for
Video 8 digital
compression.



More sophisticated companding systems usually divide the input data stream into frames, typically between 1 and 32 ms long. In a 16-bit system, the values for each sample can range from $-32,768$ to $+32,767$. However, it is rare for the sound in one frame to occupy the full range of sample values. If it does not, each sample can be expressed by a shorter word, combined with a gain or scaling factor applied to the complete frame. In other words, the levels are expressed in floating-point form, but with the restriction that the mantissa must have the same value throughout a frame. The NICAM systems, as used in stereo television, use this technique to reduce 14-bit words to 10-bit words, plus a 3-bit mantissa for each frame of 32 words.

There are two real problems with such systems. A frame with a single high sample value can cause low-level audio to be coarsely quantized. This is not usually audible if the frame length is short, typically not more than 1 millisecond. The greater problem is that they do not provide the large compression ratios we would like to see for many applications.

- A typical audio spectrum is made up of many discrete frequencies. With wide subbands, most or all subbands will contain one or more components, and so have to be encoded. With narrower bands, more subbands will fall within gaps in the spectrum and contain no components. These bands do not have to be encoded at all.
- The number of bits needed to encode a subband depends on the degree of masking. We must always consider the worst case, where the stimulus is at the top edge of the band, because of the steep slope of the masking curve below the stimulus frequency. For this reason, narrow bands mask much higher noise levels and can be encoded with fewer bits.

Band-splitting filters are an efficient means of obtaining a reasonable number of subbands, such as the 32 used by many subband encoders. Beyond this, however, use of filters becomes very cumbersome, and a frequency transform is used to analyze the signal. Very large numbers of subbands may be obtained by this method. A 256-subband transform filter has approximately the same complexity as a conventional filter for 32 subbands.

There is a drawback to transform filters. Filters with high-frequency resolution have poor temporal resolution, and this affects the coding of frames that include transients. If the full coding gain is taken, the resulting quantization noise is present throughout the frame. A transient in the frame will probably mask this noise after the transient, but there is very little "negative" temporal masking; thus, the noise may be heard at the beginning of the frame.

To counter this effect, some systems use transient detection and switch to a lower number of bands for frames that include a transient. A correctly designed transform filter bank may be switched quite easily without any significant increase in complexity.

Example Compression Systems

Audio Compression in MPEG

MPEG has standardized on compression schemes based on the MUSICAM system developed by Philips. Three coding layers are defined for MPEG-1, but Layer II is by far the most used.

Layer II supports audio sampled at 32 kHz, 44.1 kHz, and 48 kHz. It uses 24-ms frames (with 48-kHz sampling) and codes 32 equal subbands. For each subband, a 6-bit scale factor is used, giving a range of 120 dB. The scale factor can operate over the complete frame, but can be changed on 8-ms boundaries if required.

This system uses forward adaptive bit allocation and fractional bit quantization. It can code mono or stereo at bit rates of 32 to 384 kb/s. Layer II has been adopted for several direct broadcast satellite (DBS) systems and by the European Digital Video Broadcasting (DVB) system. It is also the standard for multimedia applications such as CD-ROMs.

MPEG-2 added some flexibility to MPEG-1 audio. Half-sample rates were added (16 kHz, 22.05 kHz, and 24 kHz) and provision was made for a multichannel extension. This approach provides backward compatibility with MPEG-1. If multichannel sound is required, a stereo mix must be derived and transmitted as Layer II audio. The additional information is sent in an enhancement stream. Multichannel decoders accept both streams, and remultiplex for the required number of channels. Simple decoders receive only the stereo bit stream and ignore the enhancement layer. Total bit rates of up to 1 Mbit/s are permitted for highest-quality multichannel.

The most well-known audio compression system is, of course, MP3—the designation for MPEG Audio Layer III. This uses more sophisticated algorithms to achieve greater compression efficiency, and is claimed to offer CD quality at about 64 kbits/s per channel. The availability of many thousands of MP3 compressed files on the Internet has done more to change the way we think about entertainment content than any other single development. It is far from certain how the copyright issues will be solved, and who will win the battle for control of digital entertainment distribution, but it is certain that the new techniques are here to stay.

Audio Compression for ATSC

During the process leading to the U.S. ATSC standard, it was decided to use MPEG-2 (MP@HL) for video coding. However, no agreement could be reached on the audio compression system to be used.

One proponent offered a system where six channels were individually compressed, but the main contenders were Philips with an enhanced MUSICAM system and Dolby with its AC-3 system. Both systems offered "5.1" channels. This means five full-bandwidth channels for

front left, front center, front right, left surround, and right surround. An additional channel has only narrow bandwidth and is intended to drive subwoofers, chair-shakers, and other very low-frequency devices. It is also known as the *low-frequency effects* (LFE) channel.

In a bitterly fought contest, expert listeners compared the two systems under carefully controlled studio conditions. AC-3 was selected, on the basis of its slightly superior performance on certain vibraphone passages. Philips contested the decision on the grounds that the MUSICAM system had not been performing correctly at the tests, but the decision remained. In fairness it should be said that when either system is working optimally, expert listeners cannot distinguish the results from the original recording, although Dolby claims this performance level at a lower bit rate. AC-3 supports bit rates from 32 to 640 kbits/s; the ATSC tests were performed at the intended transmission rate of 384 kbits/s (the maximum permissible rate has now been increased to 448 kbits/s, the same maximum as specified for DVDs). This represents a compression ratio of about 10:1, making the quality a remarkable achievement.

AC-3 is a transform system, switching between 256 subbands for normal operation and 128 subbands for transients. It supports the usual sampling rates of 32 kHz, 44.1 kHz, and 48 kHz, and uses 32-ms framing at 48 kHz. It uses a 4.5-bit scale factor over a range of 144 dB and forward/backward adaptive bit allocation.

Dolby incorporated a number of interesting features in AC-3, to be known as Dolby Digital in the commercial world. Although the system can encode up to 5.1 channels, the decoder can produce whatever downmix is appropriate for the receiver capabilities. Decoders can downmix to mono or to stereo or, of course, support full 5.1-channel output. In addition, each Dolby Digital (DD) decoder incorporates a Dolby Surround Pro Logic (DSPL) decoder to support the most common surround-sound installations. In fact, Dolby Digital includes a specific pass-through mode for DSPL encoded audio. The two DSPL signals are carried on the left and right front channels of the 5.1 system and passed directly to the DSPL decoder.

It is important to recognize the existence of this mode. Many videotapes, particularly film-tape transfers, have DSPL audio. DSPL is an analog compression scheme that carries center and surround information coded into the left and right channels; it is a four- to two- to four-channel system. It might be assumed that a DSPL signal could be decoded to four channels and fed to a DD system, perhaps feeding the surround signal to both left and right rear channels. This would

work if there were five loudspeakers at the receiver, but there is a hidden trap.

DSPL is an analog system where the matrixing can never be perfect, so the system is designed with this in mind. Specifically, the system is designed to avoid sensitivity to phase variations in high frequencies in the surround channel. Also, DSPL is not a true surround-sound system in that it does not attempt to place sounds accurately in a 360° stage. It provides accurate location in front of the listener, plus a general (and very effective) ambiance from behind. As a part of this process the DSPL decoder low-pass-filters and delays the derived ambiance signal. Provided the four signals are then conveyed transparently to four loudspeakers, all will be well.

However, if the DSPL signal is decoded and fed to a DD system and decoded at a receiver with only (two-channel) stereo capability, the DD decoder downmixes for the stereo output. If the signals it mixes include the filtered and delayed surround from a DSPL decoder, the downmix will be quite wrong and objectionable.

To avoid this problem, DD provides the DSPL mode described above. The two DSPL-encoded channels are fed to the left and right front channels of the DD system, and the "message" that this is DSPL is passed with the bit stream to the decoder, which passes the two channels directly to the incorporated DSPL decoder. No downmix is required for stereo, and the two DSPL channels can be combined for mono.

Dolby Digital also addresses the perennial problem of loudness variation between programs and between channels. The system uses dialog level to normalize receiver volume; the bit stream carries an indication of the dialog level and the decoder gain is varied to keep this essentially constant.

Finally, DD allows for adjustable dynamic range at the decoder. Wide dynamic range is perfect for home theater environments, but quite unsuitable for a portable television in a noisy kitchen. The decoder can reduce dynamic range as determined by the receiver manufacturer—or according to the preferences of the listener if user controls are provided—with separate control of compression above and below the nominal dialog level.

Some issues surrounding the use of DD for digital television have not yet been resolved. One is the metadata (descriptive data) that should accompany the audio to the encoder. The encoder must know if its input is DSPL, so as to invoke the pass-through mode. It also needs to know the nominal dialog level and the dynamic range setting in order to convey these to the receiver. This presents a practical issue

in a television plant where no obvious mechanism exists for routing, recording, and distributing the metadata with the audio.

Initially most potential users assumed that DD would be used for network distribution as well as final transmission to the home. However, fades, voiceovers, and the like cannot be performed on the coded signal, and such operations in a local TV station require decoding and re-encoding. As with any compression system, this results in quality loss. It has been suggested that DD coded at a higher rate than that used for transmission could be used for distribution to provide the necessary quality headroom. However, it is possible that a different coding system, Dolby-E, may be preferable for distribution; this is discussed briefly in Chapter 22 under Mezzanine Compression Systems. As with many aspects of digital television, implementation is very slow, and new decisions are being made all the time.

CHAPTER **21**

Streaming Media

Introduction

Streaming is a technique used to deliver multimedia content, often including video and audio, usually compressed. It is used on networks, including private intranets and the Internet. Streaming brings together characteristics of two fundamentally different delivery systems.

In the world of digital studio installations, both video and audio, connections between pieces of equipment are usually made by dedicated circuits. These circuits carry the bitstream, usually uncompressed, in real time with no significant transit delay. They are deterministic and isochronous—bits go into the circuit at a fixed rate and bits come out at the same fixed rate. For every bit that goes in, one comes out. The data is received by hardware elements designed specifically for the communication circuit being used, and every usable element, be it an audio sample or a video pixel, is processed as it arrives.

In this scenario there is really no concept of a beginning or end; the bitstream may last for a second, or for many years. The transmitting device sends the information it has; the receiving device accepts the information as it receives it. Even when there is no real information to send—if the video is black or the audio silent—as long as the equipment is powered, it will faithfully send the codes for black or silence, and they will be diligently received.

In the world of computers, the usual method of transferring information is a file transfer. One application creates a file and causes it to be stored on some medium. Another application may access the file from its original storage location, or some separate mechanism may cause the file to be transferred to a different storage location before the second application accesses it. The transfer mechanism may move a physical medium like a floppy disk ("sneakernet"), but today it is more likely that the file is transferred over some form of network. The file may be transferred quickly or slowly, perhaps in many parts, and the parts may not arrive in the same order as they were sent. It is the job of the transfer mechanism to ensure that when the job is finished, the received file is the same as the original.

Whatever the mechanism, the process is essentially serial. The file is created, then moved, then accessed. If there is no information to send, no file is created. No access to a file is possible until the transfer is complete.

This is the mechanism used, for example, when downloading an MP3 file from a music distributor. The file is transferred over the Internet from the server to the user's hard drive. Only when the transfer is complete can the file be accessed by an MP3 player.

The first system described above is really a very specialized form of streaming, but the term today is generally used to refer to transmission of content over a network. What makes streamed data special is that the content may be accessed and used without waiting for the completion of a file transfer. An example of streaming is the sample audio clips offered by on-line vendors of CDs. Many offer the ability to hear a few seconds of one or more tracks before making a purchase decision.

Streaming in this environment is not the same as the isochronous studio interface. We must assume a connection that supports a rate at least close to the data rate of the clip being played. If the clip is encoded at 384 kbits/s and we attempt to stream over a 20 kbits/s link, the best we could offer is one second of audio followed by a 20-second wait, then one more second of audio. In this case, waiting for a complete file transfer and playing the clip from the hard disk is the only practical solution.

But even if the average link speed is sufficient to support approximately real-time transfer, the transfer rate will rarely be constant. Delivery via the Internet generally involves many links and many routers. Transmission is usually "bursty" and packets may not be delivered in the order that they were transmitted. Clearly buffering is needed to permit an acceptable presentation. An intelligent player application will adjust the buffer time according to the characteristics of the incoming data; if the buffer empties (resulting in a break in presentation) the buffer size will be increased to make future interruptions less likely.

Applications for Streaming Media

The usefulness of streaming media is totally dependent on the availability of adequate bandwidth. The comments that follow are based on the environment as seen late in the year 2000. I make some pessimistic predictions below, and I hope I shall be proved wrong, as most commentators are! Time will tell.

In corporate intranets this may not be an issue: 100 Mbits/s Ethernet is starting to replace 10 Mbits/s, star network topology is rapidly replacing loops, and switches are replacing hubs. All of these factors help to increase the bandwidth available to a user and to reduce bottlenecks in the network. Streaming video over corporate networks is viable today, and the situation is continually improving.

Away from this environment the situation is much less encouraging. Audio streaming is useful on the Internet. The standard home connection is a 56 kbits/s modem, often achieving about 40 kbits/s. Modern techniques of audio compression can offer acceptable speech quality at 10 kbits/s or less and reasonable music quality from about 20 kbits/s. The standard connection, and the Internet itself, offer sufficient overhead for these rates to be supported quite consistently.

As mentioned above, e-commerce vendors can offer music samples that genuinely improve the shopping experience. Congressional hearings and shareholder meetings are now routinely "broadcast" over the Internet.

Video is a very different story and has, to date, achieved little more than novelty value over the Internet. High-quality, full-screen, full-refresh-rate video still requires megabits per second. Improvements in compression technology will help, but there is no way we will be able to deliver serious video in a few hundred kilobits per second. While we are confined to connections with even lower bandwidths, Internet streaming video will likely remain postage-stamp sized, and/or have very a low refresh rate.

There are useful applications for streaming video. As mentioned above, corporate intranets can provide a suitable infrastructure. Aside from this, video conferencing and distance learning are attractive applications that can deliver real benefits even today, but the value is greatest where the video is a supplement to other content (audio, faxed documents, prepublished hard copy material, electronic slide presentations, etc.). Generally videoconferencing and distance learning use leased network connections or dial-up ISDN, offering at least 128 kbits/s. Even at this, quality is low, update is slow, and either the video is delayed with respect to the audio, or both are delayed, making conversation difficult.

We hear talk of the "bandwidth explosion." Will the situation improve soon? Personally I think not, even though there is a great deal of unused fiber on long-distance routes. High-speed home connections are available in the form of DSL and cable modems. These certainly offer an improved experience, but mainly because there are so few of them. There is little evidence that the switching infrastructure supporting these connections is growing at anything like the necessary pace. More and more Web sites are offering content that requires more bandwidth. I see an explosion of bandwidth demand and creeping infrastructure bandwidth growth.

The spread of multicasting technology will certainly help with applications such as "live" broadcast. Today, almost all Internet stream-

ing relies on unicast operation—the server must deliver a separate bitstream to each individual user, addressed appropriately, even if thousands of users are watching a live event. When more multicast technology is available, servers will be able to deliver a single bitstream to which multiple users may subscribe.

Standards for Streaming Media

The first standard to achieve widespread adoption was H.261, an ITU Standard approved in 1990. It is intended for use on ISDN circuits and uses multiples of 64 kbits/s. It supports the common image format (CIF) at 352×276 pixels and the quarter-size QCIF at 176×144 pixels. It uses a combination of motion compensation, DCT, and VLC, and can code at rates up to 30 frames/s. Unlike MPEG, H.261 and other streaming media standards permit frame dropping as a means to control bit rate. It is very common in video conferencing systems to see the image freeze when some rapid motion, such as a pan, occurs.

H.263 followed in 1996. Initially, this standard was intended for very low bit rates only (up to 64 kbits/s), but this restriction was removed and the standard is likely to replace H.261. It employs improved coding algorithms and supports QCIF and SQCIF (approximately half-QCIF). As an option, implementations can support higher resolutions up to 16 CIF at $1,408 \times 1,152$ pixels.

More recently, the H.323 series of standards provides higher quality over corporate intranets and to institutions with second-generation Internet connections. Operation is typically in the hundreds of kilobits/s range. The higher quality offered by these standards and bandwidths is particularly attractive to academic and medical conferencing.

In the world of personal computers there is intense rivalry among three companies, Apple, RealNetworks, and Microsoft. The original format, and still very popular, is *Quicktime* by Apple. This format supports a wide range of bit rates and compression schemes—nowadays almost anything can be made into a Quicktime movie. On the CD included with this book you will find commercial stock footage at full resolution, and low bit rate preview versions of the same clips; both are supplied as Quicktime movies. There is a free version of the Quicktime player, but Apple does not seem to offer any free version of the encoder.

RealNetworks range of products including *RealAudio* and *RealVideo* have probably shown the fastest growth in recent years as many web sites have added audio and video elements. As mentioned previously, we now expect to hear audio samples from CDs before we buy them online. The RealNetworks solutions have proven very attractive at the low bit rates available on today's Internet. RealNetworks offer a free basic player, and a trial version of the encoder is also available for download without charge. More sophisticated players, and fully functional encoders, are available for a price.

The most recent major contender is Microsoft Corporation. It has mounted a major effort to redefine streaming media with the *Advanced Streaming Format* (ASF), offering end-to-end solutions. Microsoft offers a complete range of tools including players, encoders, servers, and supplementary tools at its *Windows Media* site. Microsoft's new Media Player, available at this site and included with the latest operating systems, supports a wide range of coding formats (except Real), and the latest encoder supports MPEG-4 (Version 1 at the time of writing). The CD includes sample Windows Media files, at audio/video bit rates ranging from 28 kbits/s to 2Mbits/s. Microsoft's determination to become a substantial player in the field of streaming media may be deduced from the fact that (at the time of writing) *all* of these tools are available without charge.

Some of the tools mentioned above will be found on the CD-ROM included with this book; others may be obtained from the respective Web sites. In all cases, it would be wise to check the Web sites for the latest versions of these products.

Microsoft Corporation has mounted a major effort to redefine streaming media, with *Advanced Streaming Format* (ASF) offering end-to-end solutions. Microsoft offers a full range of free tools, including encoders, players, and servers, plus a wealth of information at its Web site. Microsoft's new Media Player, available at this site, and included with the latest operating systems, supports a vast range of coding formats (except Real), and the latest encoder supports MPEG-4 (Version 1 at the time of writing).

Some of these tools are on the CD-ROM included with this book, but check out the Web sites for latest versions.

 Update 2003

The technology of streaming media has come a long way since this chapter was written. MPEG-4 implementations have been tried, but largely abandoned as proprietary technology outperformed the MPEG standard. Now things are changing again. The JVT codec, discussed in Chapter 12, is expected to perform well in streaming applications. This codec does require considerable processing power, even for the decoder, but it will likely find be used soon in streaming applications if the licensing terms are attractive. Microsoft is making another bid for the streaming market with Windows Media 9 (see Chapter 13) with a simpler codec, performance believed to be comparable to JVT, and aggressive licensing terms.

The good news is that McGraw-Hill has published *Streaming Media Demystified* by my friend Michael Topic (see bibliography). This should provide an excellent source of information for those who wish to know (much) more about the subject.



YOUR ONE-STOP GUIDE TO THE ART AND SCIENCE OF DIGITAL VIDEO COMPRESSION!

CD-ROM WITH ALGORITHMS, ENCODERS, AND PRACTICE CLIPS

Without video compression, video streaming—one of the Internet's great innovations—would be nothing more than a pipe dream. That's why if digitizing video imagery is part of your work or hobby, here's an info-packed volume you need handy at all times: McGraw-Hill's *Digital Video Compression*.

Written by a renowned expert in digital technology, *Digital Video Compression* offers comprehensive coverage on a vast range of digital media technologies and standards. It delivers everything you need to understand the techniques, tricks, and confusing protocols of compression, as well as:

- Fundamentals of compression
- MPEG-2 and MPEG-4 compression standards
- JVT / H.264 / MPEG-4 Part 10—learn the details of the just-released compression standard
- And much more!

Plus, you get an invaluable CD-ROM of compression tools, encoders, and video clips for practice. With *Digital Video Compression*, you're sure to stay out in front of this exploding technology!

COVERAGE INCLUDES

- What Is Compression?
 - An Introduction to Images
 - Entropy Coding
 - Protective Coding
 - Transforms
 - Quantization
 - JPEG
- Motion Compensation
 - MPEG-1
 - MPEG-2
 - MPEG-4
 - JVT / H.264 / MPEG-4 Part 10
- MPEG-7 and MPEG-21
- Pro-MPEG and MPEG Operating Ranges
- DV Compression
 - Wavelets
 - JPEG2000
- Audio Compression
- Streaming Media

The McGraw-Hill Companies

P/N 142494-B PART OF

ISBN 0-07-142487-3



9 780071 424875



54495



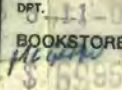
6 39785 150689 8

Visit McGraw-Hill Education at:
www.books.mcgraw-hill.com

Cover design: The Gazillion G

\$44.95 U.S.A. | £29.99 U.K.

Web Graphics/Video



Ex. 1020A

LIBRARY | CATALOG

◀ 1 of 1 ▶

BOOK

Digital video compression

[Full Record](#) [MARC Tags](#)

Personal name

Symes, Peter (Peter D.)

Main title

Digital video compression / Peter Symes.

Published/Created

New York : McGraw-Hill, c2004.

[Request this item](#)[LC Find It](#)

Links

Links

Contributor biographical information <http://www.loc.gov/catdir/bios/mh042/2003065185.html>

Table of contents <http://www.loc.gov/catdir/toc/mh041/2003065185.html>

More Information

LCCN Permalink<https://lccn.loc.gov/2003065185>**Description**

xvii, 394 p. : ill. ; 24 cm. + 1 CD-ROM (4 3/4 in.)

ISBN	0071424873
LC classification	TK6680.5 .S93 2004
Related names	Symes, Peter (Peter D.) Video compression demystified
LC Subjects	Video compression. Digital video. MPEG (Video coding standard) JPEG (Image coding standard)
Browse by shelf order	TK6680.5
Notes	Revision and extension of: Video compression demystified .
LCCN	2003065185
Dewey class no.	621.388/33
Invalid ISBN	0071424946
Type of material	Book

Item Availability



CALL NUMBER	TK6680.5 .S93 2004 FT MEADE Copy 2
Request in	Machine Readable Collections - STORED OFFSITE
Status	Not Charged

CALL NUMBER	TK6680.5 .S93 2004 CABIN BRANCH Copy 1
Request in	Machine Readable Collections - STORED OFFSITE
Status	Not Charged

LIBRARY | CATALOG



1 of 1

BOOK

Digital video compression

Full Record MARC Tags

000	01415cam a22003254a 4500
001	13342335
005	20041215092348.0
008	030912s2004 nyua 001 0 eng
906	— a 7 b dbc c orignew d 1 e ocip f 20 g y-gencatlg
925	0_ a acquire b 2 shelf copies x policy default
955	— a cc20 2003-09-12 to ASCD c j105 2003-09-15 to subj. d j107 2003-09-16 to SL e j112 2003-09-24 to Dewey a aa07 2003-09-26 a ps13 2004-05-19 bk rec'd, to CIP ver.; f jg02 2004-06-02; g jg07 2004-06-15 to BCCD, copy 1; a jg07 2004-12-15 to BCCD, copy 2
010	— a 2003065185
020	— a 0071424873
020	— z 0071424946
040	— a DLC c DLC d DLC
042	— a ooc
050	00 a TK6680 .5 b .S93 2004
082	00 a 621.388/33 2 22
100	1_ a Symes, Peter q (Peter D.)
245	10 a Digital video compression / c Peter Symes.
260	— a New York : b McGraw-Hill, c c2004.
300	— a xviii, 394 p. : b ill. ; c 24 cm. + e 1 CD-ROM (4 3/4 in.)
500	— a Revision and extension of. Video compression demystified.
650	_0 a Video compression.
650	_0 a Digital video.

650	_0	a MPEG (Video coding standard)
650	_0	a JPEG (Image coding standard)
700	1_	a Symes, Peter q (Peter D.) t Video compression demystified
856	42	3 Contributor biographical information u http://www.loc.gov/catdir/bios/mh042/2003065185.html
856	41	3 Table of contents u http://www.loc.gov/catdir/toc/mh041/2003065185.html

Request this Item

 LC Find It

Item Availability

>

CALL NUMBER	TK6680.5 .S93 2004 FT MEADE Copy 2
Request in	Machine Readable Collections - STORED OFFSITE
Status	Not Charged

CALL NUMBER	TK6680.5 .S93 2004 CABIN BRANCH Copy 1
Request in	Machine Readable Collections - STORED OFFSITE
Status	Not Charged



THE MPEG-4 BOOK

Edited by Fernando Pereira and Touradj Ebrahimi

IMSC Press Multimedia Series/Andrew Tescher, Series Editor



Google Exhibit No. 1021
Page 001

Google Exhibit No. 1063
Page 0765

TK
06305
P47
2102

The MPEG-4 Book

Fernando Pereira
Touradj Ebrahimi
IMSC Press



Prentice Hall PTR
Upper Saddle River, NJ 07458
www.phptr.com

Google Exhibit No. 1021
Page 002

Google Exhibit No. 1063
Page 0766

Library of Congress Cataloging-in-Publication Data

Pereira, Fernando C. N.
The MPEG-4 book / Fernando Pereira, Touradj Ebrahimi
p. cm.-- (IMSC Press multimedia series)
Includes bibliographical references and index.
ISBN 0-13-061621-4
1. MPEG (Video coding standard). 2. Video compression—Standards.
I. Ebrahimi, Touradj. II. Title. III. Prentice Hall IMSC multimedia series.

TK6680.5 .P47 2002
621.388—dc21

2002070433

Editorial/production supervision: *BooksCraft, Inc., Indianapolis, IN*
Acquisitions editor: *Bernard Goodwin*
Editorial assistant: *Michelle Vincenti*
Marketing manager: *Dan DePasquale*
Manufacturing manager: *Alexis Heydt-Long*
Cover design director: *Jerry Votta*
Cover designer: *Anthony Gemmellaro*
Full-service production manager: *Anne Garcia*



© 2002 Pearson Education, Inc.
Publishing as Prentice Hall PTR
Upper Saddle River, New Jersey 07458

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

For information regarding corporate and government bulk discounts please contact:
Corporate and Government Sales (800) 382-3419 or corpsales@pearsontechgroup.com

Company and product names mentioned herein are the trademarks or registered trademarks of their respective owners.

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.

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

ISBN 0-13-061621-4

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd.
Pearson Education Canada, Ltd.
Pearson Educación de México, S.A. de C.V.
Pearson Education—Japan
Pearson Education Malaysia, Pte. Ltd.

Contents

Foreword	xix
Preface	xxiii
Abbreviations	xxvii
1 Context, Objectives, and Process	1
1.1 MPEG-4 Objectives	4
1.1.1 Functionalities	5
1.1.2 Requirements	7
1.1.3 Tools	10
1.1.4 Applications	12
1.2 Formal Standardization Process	13
1.3 MPEG Modus Operandi	15
1.3.1 Mission	16
1.3.2 Principles	17
1.3.3 Standards Development Approach	18
1.4 MPEG-4 Standard Organization	21
1.5 MPEG-4 Schedule	25
1.6 MPEG-4 Industry Forum	30
1.7 Summary	31
1.8 References	32

vii

2 MPEG-4 Overview	37
2.1 Design Goals.....	38
2.2 An End-to-End Walkthrough.....	40
2.3 Terminal Architecture.....	43
2.4 MPEG-4 Tools.....	47
2.4.1 Systems Tools	47
2.4.2 Visual Tools.....	48
2.4.3 Audio Tools.....	50
2.4.4 DMIF Tools	51
2.4.5 Other MPEG-4 Tools	51
2.4.6 Profiles and Levels	52
2.5 MPEG-4 and Other Multimedia Standards	52
2.6 MPEG-4 Applications	56
2.6.1 Multimedia Portals.....	56
2.6.2 Interactive Broadcasting.....	57
2.6.3 Multimedia Conferencing and Communities	59
2.7 Summary	60
2.8 References	61
3 Object Description and Synchronization	65
3.1 Object Descriptors: Entry Points to MPEG-4 Content.....	65
3.1.1 Syntactic Description Language	66
3.1.2 Object Description	70
3.1.3 Stream Description.....	72
3.1.4 Stream Relationship Description.....	76
3.1.5 Content Complexity Description.....	77
3.1.6 Streaming ODs	79
3.1.7 Linking a Scene to Its Media Streams	82
3.1.8 MPEG-4 Content Access Procedure.....	85
3.2 Semantic Description and Access Management.....	86
3.2.1 Object Content Information: Meta Information About Objects	86
3.2.2 Intellectual Property Management and Protection	89
3.3 Timing Model and Synchronization of Streams	90
3.3.1 Modeling Time	90
3.3.2 Time Stamps and Access Units.....	92
3.3.3 Packetizing Streams: The Sync Layer.....	93
3.3.4 Distributed Content, Time Bases, and OCR Streams	96
3.3.5 Media Time	98
3.3.6 System Decoder Model	98
3.4 Summary	100
3.5 References	101
4 BIFS: Scene Description	103
4.1 Basics of BIFS	104
4.1.1 Scene and Nodes	104
4.1.2 Fields and ROUTEs.....	106

4.1.3 Node Types	108
4.1.4 Subscenes and Hyperlinks	109
4.1.5 Scene Changes	109
4.1.6 Scene Rendering	112
4.1.7 Binary Encoding	113
4.1.8 Quantization	114
4.2 Basic BIFS Features by Example	114
4.2.1 Trivial Scene	114
4.2.2 Movie with Subtitles	117
4.2.3 Icons and Buttons	119
4.2.4 Slides and Transitions	121
4.2.5 Simple 3D Scene	124
4.2.6 Magnifying Glass	128
4.3 Advanced BIFS Features	129
4.3.1 Scripting	130
4.3.2 Encapsulation and Reuse	131
4.3.3 Text Layout	134
4.4 A Peek Ahead on BIFS	136
4.4.1 Media Nodes	136
4.4.2 New Sensors	137
4.4.3 FlexTime	138
4.4.4 ServerCommand	139
4.5 Profiles	139
4.6 All BIFS Nodes	140
4.7 Summary	146
4.8 References	147
5 MPEG-J: MPEG-4 and Java	149
5.1 MPEG-J Architecture	154
5.1.1 MPEGlets	156
5.1.2 Delivery	158
5.1.3 Security	159
5.2 MPEG-J APIs	160
5.2.1 Terminal APIs	163
5.2.2 Scene APIs	164
5.2.3 Resource APIs	173
5.2.4 Decoder APIs	176
5.2.5 Network APIs	178
5.2.6 Service Information and Section Filtering APIs	180
5.2.7 MPEG-J Profiles	180
5.3 Application Scenarios	181
5.3.1 Adaptive Rich Media Content for Wireless Devices	181
5.3.2 Enhanced Interactive Electronic Program Guide (EPG)	181
5.3.3 Enriched Interactive Digital Television	182
5.3.4 Content Personalization	182

5.4 Reference Software	182
5.5 Summary	183
5.6 References	185
6 Extensible MPEG-4 Textual Format.....	187
6.1 Objectives	188
6.2 Cross-Standard Interoperability	189
6.3 XMT Two-Tier Architecture	190
6.4 XMT-Ω Format	193
6.4.1 Reusing SMIL in XMT-Ω.....	194
6.4.2 Extensible Media (xMedia) Objects	195
6.4.3 Timing and Synchronization.....	198
6.4.4 Time Manipulations	202
6.4.5 Animation.....	203
6.4.6 Spatial Layout	205
6.4.7 XMT-Ω Examples.....	205
6.5 XMT-A Format.....	210
6.5.1 Document Structure	210
6.5.2 Timing	211
6.5.3 Scene Description	212
6.5.4 Object Descriptor Framework.....	219
6.5.5 Deterministic Mapping of XMT-A	222
6.5.6 Interoperability with X3D.....	223
6.6 Summary	224
6.7 References	224
7 Transporting and Storing MPEG-4 Content	227
7.1 Delivery Framework	229
7.1.1 DMIF-Application Interface.....	231
7.1.2 DMIF Network Interface.....	238
7.1.3 Existent Signaling Protocols.....	246
7.2 FlexMux Tool	248
7.2.1 Timing of a FlexMux Stream	251
7.3 MPEG-4 File Format.....	253
7.3.1 Temporal Structure: Tracks and Streams, Time and Durations	254
7.3.2 Physical Structure: Atoms and Containers, Offsets and Pointers.....	257
7.3.3 MPEG-4 Systems Concepts in MP4.....	259
7.3.4 MPEG-4 Track Types and Storage	259
7.3.5 Hinting	260
7.3.6 Atoms.....	262
7.3.7 Random Access	265
7.3.8 An MP4 Example.....	266
7.3.9 Summary of MP4	268
7.4 Transporting MPEG-4 over MPEG-2.....	268
7.4.1 Brief Introduction to MPEG-2 Systems.....	269
7.4.2 Transport of MPEG-4 Elementary Streams over MPEG-2 Systems	274

7.4.3 Transport of MPEG-4 Scenes over MPEG-2 Systems	275
7.5 Transporting MPEG-4 over IP	280
7.5.1 Brief Introduction to Streaming over IP	281
7.5.2 Transport of Elementary Streams over IP	283
7.5.3 Transport of SL-Packetized Streams over IP	285
7.5.4 FlexMux Streams over IP and Timing Models	288
7.6 Summary	289
7.7 References	289
8 Natural Video Coding	293
8.1 General Overview	294
8.1.1 Functionalities and Application Scenarios	294
8.1.2 Basic Principles	295
8.2 Coding of Rectangular Video Objects	300
8.2.1 Overview	300
8.2.2 New Motion-Compensation Tools	303
8.2.3 New Texture Coding Tools	311
8.3 Coding of Arbitrarily Shaped Video Objects	318
8.3.1 Binary Shape Coding	318
8.3.2 Gray-Level Shape Coding	323
8.3.3 Coding of Boundary Macroblocks	324
8.4 Scalable Video Coding	330
8.4.1 Spatial Scalability	332
8.4.2 Temporal Scalability	337
8.4.3 SNR Fine Granularity Scalability	340
8.5 Special Video Coding Tools	343
8.5.1 Interlaced Coding	344
8.5.2 Error-Resilient Coding	345
8.5.3 Reduced Resolution Coding	355
8.5.4 Sprite Coding	356
8.5.5 Short Video Header Mode	359
8.5.6 Texture Coding for High-Quality Applications	359
8.6 Visual Texture Coding	361
8.6.1 VTC Tools	363
8.6.2 Wavelet Coding	364
8.6.3 Shape-Adaptive Wavelet Coding	368
8.6.4 Spatial and Quality Scalability	369
8.6.5 Bitstream Packetization	371
8.6.6 Tiling	374
8.7 Summary	375
8.8 References	376
9 Visual SNHC Tools	383
9.1 SNHC Overview	384
9.1.1 VRML, X3D: Why Is SNHC Needed?	385
9.1.2 SNHC Visual Tools	388

9.2 Face and Body Animation.....	389
9.2.1 Overview	389
9.2.2 Default Facial Expression and Body Posture	392
9.2.3 FBA Object.....	393
9.2.4 FDP and BDP Listing and Coding.....	396
9.2.5 FAP and BAP Listing and Coding.....	403
9.2.6 FBA and Text-to-Speech Interface	410
9.3 2D Mesh Coding.....	411
9.3.1 2D Mesh Object.....	412
9.3.2 Coding Scheme.....	413
9.3.3 Example	420
9.4 3D Mesh Coding.....	421
9.4.1 3D Mesh Object.....	422
9.4.2 Coding Scheme.....	425
9.4.3 Examples.....	437
9.5 View-Dependent Scalability	440
9.5.1 View-Dependent Object.....	442
9.5.2 Coding Scheme.....	444
9.5.3 Example	446
9.6 Profiles and Levels	446
9.7 Summary	447
9.8 Acknowledgments	447
9.9 References	448
10 Speech Coding.....	451
10.1 Introduction to Speech Coding	452
10.2 Overview of MPEG-4 Speech Coders	454
10.3 MPEG-4 CELP Coding	455
10.3.1 CELP Encoder.....	455
10.3.2 CELP Decoder.....	456
10.3.3 Parameter Decoding	457
10.3.4 Multipulse Excitation	458
10.3.5 Regular Pulse Excitation.....	459
10.3.6 Scalability	460
10.3.7 Silence Compression.....	461
10.4 MPEG-4 HVXC Coding.....	461
10.4.1 HVXC Encoder.....	462
10.4.2 HVXC Decoder	470
10.4.3 Variable Bit-Rate Coding	474
10.5 Error Robustness	475
10.5.1 Error-Resilient HVXC	477
10.5.2 Error-Resilient CELP	481
10.6 Summary	482
10.7 References	483

11 General Audio Coding.....	487
11.1 Introduction to Time/Frequency Audio Coding.....	488
11.2 MPEG-2 Advanced Audio Coding	490
11.2.1 Coder Overview	491
11.2.2 Gain Control	491
11.2.3 Filterbank	493
11.2.4 Quantization.....	495
11.2.5 Noiseless Coding.....	497
11.2.6 Temporal Noise Shaping.....	499
11.2.7 Prediction.....	502
11.2.8 Joint Stereo Coding.....	504
11.2.9 Bitstream Multiplexing.....	507
11.2.10 Other Aspects	508
11.3 MPEG-4 Additions to AAC	508
11.3.1 Perceptual Noise Substitution.....	509
11.3.2 Long-Term Prediction	511
11.3.3 TwinVQ	513
11.3.4 Low-Delay AAC (AAC-LD).....	515
11.3.5 Error Robustness.....	517
11.4 MPEG-4 Scalable Audio Coding	518
11.4.1 Large-Step Scalable Audio Coding.....	519
11.4.2 Bit-Sliced Arithmetic Coding.....	523
11.5 Introduction to Parametric Audio Coding	525
11.5.1 Source and Perceptual Models.....	526
11.5.2 Parametric Encoding and Decoding Concepts	528
11.6 MPEG-4 HILN Parametric Audio Coding.....	530
11.6.1 HILN Parametric Audio Encoder.....	530
11.6.2 HILN Bitstream Format and Parameter Coding	535
11.6.3 HILN Parametric Audio Decoder	538
11.7 Summary.....	539
11.8 Acknowledgments	540
11.9 References	540
12 SNHC Audio and Audio Composition.....	545
12.1 Synthetic-Natural Hybrid Coding of Audio.....	546
12.2 Structured Audio Coding.....	548
12.2.1 Algorithmic Synthesis and Processing.....	549
12.2.2 Wavetable Synthesis	552
12.2.3 Interface Between Structured Audio Coding and AudioBIFS.....	554
12.2.4 Structured Audio Applications	554
12.3 Text-to-Speech Interface.....	555
12.4 Audio Composition.....	557
12.4.1 VRML Sound Model in BIFS	558
12.4.2 Other AudioBIFS Nodes	560
12.4.3 Enhanced Modeling of 3D Audio Scenes in MPEG-4	567

12.4.4 Advanced AudioBIFS for Enhanced Presentation of 3D Sound Scenes.....	569
12.5 Summary	580
12.6 References	580
13 Profiling and Conformance: Approach and Overview.....	583
13.1 Profiling and Conformance: Goals and Principles.....	584
13.2 Profiling Policy and Version Management.....	588
13.3 Overview of Profiles in MPEG-4	592
13.3.1 Visual Profiling	592
13.3.2 Audio Profiling	602
13.3.3 Graphics Profiling.....	609
13.3.4 Scene Graph Profiling	613
13.3.5 Object Descriptor Profiling.....	619
13.3.6 MPEG-J Profiling	619
13.4 Summary	620
13.5 Acknowledgements.....	620
13.6 References	620
14 Implementing the Standard: The Reference Software.....	623
14.1 Reference Software Modules	625
14.2 Systems Reference Software	625
14.3 MPEG-4 Player Architecture.....	628
14.3.1 Player Structure	631
14.4 Scene Graph	632
14.4.1 Parsing the Scene Graph.....	633
14.4.2 Rendering the Scene Graph	638
14.5 PROTOs	639
14.6 Synchronization	643
14.6.1 Sync Layer	643
14.6.2 MediaStream Objects.....	644
14.7 Object Descriptors.....	646
14.7.1 Syntactic Description Language	646
14.7.2 Parsing the OD Stream	647
14.7.3 Execution of OD Objects.....	651
14.8 Plug-Ins	652
14.8.1 DMIF Plug-Ins	652
14.8.2 Decoder Plug-Ins.....	654
14.8.3 IPMP Plug-Ins	656
14.9 2D Compositor	656
14.9.1 Using the Core Framework.....	657
14.9.2 DEF and USE Handling.....	657
14.9.3 Rendering Optimization	658
14.9.4 Synchronization	659
14.10 3D Compositor.....	660
14.10.1 Differences Between the 2D and 3D Compositors	661

14.10.2 Using the Core Framework.....	662
14.10.3 Overview of the Key Classes.....	662
14.10.4 3D Rendering Process	662
14.10.5 Support for 2D Nodes.....	664
14.10.6 User Navigation	664
14.11 Summary.....	666
14.12 References	666
15 Video Testing for Validation.....	669
15.1 General Aspects	670
15.1.1 Selection of Test Material	671
15.1.2 Selection of Test Subjects.....	671
15.1.3 Laboratory Setup.....	672
15.1.4 Test Plan.....	672
15.1.5 Training Phase	673
15.2 Test Methods.....	673
15.2.1 Single Stimulus Method.....	674
15.2.2 Double Stimulus Impairment Scale Method	675
15.2.3 Double Stimulus Continuous Quality Scale Method.....	676
15.2.4 Simultaneous Double Stimulus for Continuous Evaluation Method ...	677
15.3 Error-Resilience Test	680
15.3.1 Test Conditions.....	681
15.3.2 Test Material	684
15.3.3 Test Method and Design	684
15.3.4 Data Analysis	686
15.3.5 Test Results	686
15.4 Content-Based Coding Test.....	688
15.4.1 Test Conditions.....	688
15.4.2 Test Material	689
15.4.3 Test Method and Design	689
15.4.4 Data Analysis	690
15.4.5 Test Results	690
15.5 Coding Efficiency for Low and Medium Bit-Rate Test.....	691
15.5.1 Test Conditions.....	691
15.5.2 Test Material	692
15.5.3 Test Method and Design	692
15.5.4 Data Analysis	693
15.5.5 Test Results	693
15.6 Advanced Real-Time Simple Profile Test.....	694
15.6.1 Test Conditions.....	696
15.6.2 Test Material	699
15.6.3 Test Method and Design	700
15.6.4 Data Analysis	701
15.6.5 Test Results	702
15.7 Summary.....	704

15.8 References	705
16 Audio Testing for Validation	709
16.1 General Aspects	710
16.1.1 Selection of Test Material	710
16.1.2 Selection of Test Subjects	711
16.1.3 Laboratory Setup	712
16.1.4 Test Plan	712
16.1.5 Training Phase	712
16.2 Test Methods	713
16.2.1 Absolute Category Rating Method	713
16.2.2 Paired Comparison Method	714
16.2.3 MUSHRA Method	715
16.3 Narrowband Digital Audio Broadcasting Test	717
16.3.1 Test Conditions	718
16.3.2 Test Material	719
16.3.3 Assessment Method and Test Design	720
16.3.4 Data Analysis	721
16.3.5 Test Results	722
16.4 Audio on the Internet Test	724
16.4.1 Test Conditions	724
16.4.2 Test Material	725
16.4.3 Assessment Method and Test Design	726
16.4.4 Data Analysis	727
16.4.5 Test Results	728
16.5 Speech Communication Test	730
16.5.1 Test Conditions	730
16.5.2 Test Material	732
16.5.3 Assessment Method and Test Design	733
16.5.4 Data Analysis	734
16.5.5 Test Results	734
16.6 Version 2 Coding Efficiency Test	736
16.6.1 Test Conditions	737
16.6.2 Test Material	739
16.6.3 Assessment Method and Test Design	740
16.6.4 Data Analysis	740
16.6.5 Test Results	740
16.7 Version 2 Error-Robustness Test	744
16.7.1 Test Conditions	744
16.7.2 Test Material	747
16.7.3 Test Method and Experimental Design	747
16.7.4 Data Analysis	747
16.7.5 Test Results	747
16.8 Summary	749

16.9 References	750
A Levels for Visual Profiles.....	753
A.1 Video Buffering Verifier Mechanism	754
A.1.1 Video Rate Buffer Verifier Definition	756
A.1.2 Video Complexity Verifier Definition	760
A.1.3 Video Reference Memory Verifier Definition	764
A.1.4 Interaction Between the VBV, VCV, and VMV Models.....	766
A.2 Definition of Levels for Video Profiles.....	767
A.3 Definition of Levels for Synthetic Profiles	767
A.3.1 Scalable Texture Profile	775
A.3.2 Simple Face Animation Profile	775
A.3.3 Simple FBA Profile.....	775
A.3.4 Advanced Core and Advanced Scalable Texture Profiles.....	776
A.4 Definition of Levels for Synthetic and Natural Hybrid Profiles.....	776
A.4.1 Basic Animated Texture Profile	776
A.4.2 Hybrid Profile.....	778
A.5 References.....	778
B Levels for Audio Profiles	781
B.1 Complexity Units	781
B.2 Definition of Levels for Audio Profiles.....	783
B.2.1 Main Profile	784
B.2.2 Scalable Profile.....	784
B.2.3 Speech Profile.....	785
B.2.4 Synthetic Profile.....	785
B.2.5 High-Quality Audio Profile	786
B.2.6 Low-delay Audio Profile	786
B.2.7 Natural Audio Profile.....	786
B.2.8 Mobile Audio Internetworking Profile.....	786
B.3 References.....	788
C Levels for Graphics Profiles	789
C.1 Simple 2D Profile	789
C.2 Simple 2D + Text Profile.....	790
C.3 Core 2D Profile	790
C.4 Advanced 2D Profile	793
C.5 References.....	795
D Levels for Scene Graph Profiles	797
D.1 Simple 2D Profile	797
D.2 Audio Profile.....	798
D.3 3D Audio Profile	801
D.4 Basic 2D Profile	802
D.5 Core 2D Profile.....	802

D.6 Advanced 2D Profile.....	806
D.7 Main 2D Profile	809
D.8 References.....	813
E MPEG-J Code Samples.....	815
E.1 Scene APIs	815
E.2 Resource and Decoder APIs	817
E.2.1 Listener Class for Decoder Events.....	819
E.2.2 Listener Class for Renderer Events	820
E.3 Network APIs	820
E.4 Section Filtering APIs	821
Index.....	823

Context, Objectives, and Process¹

by *Fernando Pereira*

Keywords: audiovisual object-based representation, audiovisual objects, MPEG-4, standardization process

“What does it mean, to see? The plain man’s answer (and Aristotle’s, too) would be, to know what is where by looking. In other words, vision is the process of discovering from images what is present in the world, and where it is.” [Marr82]

The first video coding standards and their underlying representation models mainly address the vision process by providing video representation in the form of a sequence of rectangular 2D frames, giving users a window to the real world: the television paradigm. However, the process of vision is often just the initial part of the task at hand, because typically humans need and want to see, to take actions after, to interact with the objects identified. A similar reasoning can be made regarding the process of hearing and the corresponding audio representation models [Pere99].

1. Some sections of this chapter are adapted from *Signal Processing: Image Communication*, 15 (4–5) (2000), Fernando Pereira, pp. 271–279, “MPEG-4: Why, What, How and When?”, Copyright (2000), with permission from Elsevier Science.

Although the television paradigm dominated audiovisual communications for many years, the situation has been evolving quickly in terms of the ways audiovisual content is produced, delivered, and consumed [KoPC97]. Moreover, hardware and software are becoming increasingly powerful, with microelectronic technology providing new programmable processors, opening new frontiers to the representation technologies used and to the functionalities provided.

Producing content today is easier than ever before. Digital still cameras directly storing in JPEG format have hit the mass market. Together with the first digital video cameras directly recording in MPEG-1 format, this represents a major step for consumer acceptance of digital audiovisual acquisition and compression technology. This step transforms every individual into a potential content producer, capable of creating content that can be easily distributed and published on the Internet. In addition, more content is being synthetically produced—that is, computer generated—and integrated with natural material in truly hybrid audiovisual content. The various pieces of content, digitally encoded, can be successively reused without the quality losses typical of the previous analog processes.

Whereas audiovisual information, notably the visual portion, until recently was carried only over very few networks, the trend is now toward the generalization of visual information in every single network. Moreover, the increasing mobility in telecommunications is a major trend. Mobile connections will not be limited to voice; other types of data, including real-time media, are already emerging. Because mobile telephones are replaced every two to three years, new mobile devices can finally make the decade-long promise of audiovisual communications a reality. Notice that the need for visual communication is much stronger when people are not at home, and so have something to show besides the usual living room—for example, the nice beach where they are vacationing. This reinforces the relevance of audiovisual mobile communications.

The explosion of the World Wide Web and the acceptance of its interactive mode of operation have clearly shown that the traditional television paradigm will no longer suffice for audiovisual services. Users want to have access to audio and video as they now have access to text and graphics. This requires moving pictures and audio of acceptable quality at low bit rates on the Web, and Web-type interactivity with live content. It will be possible to activate relationships between users (in a potentially virtual world) through hyperlinking—the Web paradigm—and to experience interactive immersion in natural and virtual environments—the games paradigm.

As many of the emerging audiovisual applications demanded interworking, the need to develop an open and timely international standard addressing the needs mentioned above became evident. In 1993, MPEG (Moving Picture Experts Group) [MPEG] launched the MPEG-4 project, later formally called *Coding of Audio-Visual Objects*, to address (among other things) the requirements associated with the new applications resulting from these trends [N1177, N4319, N4505].

The need for any standard comes from an essential requirement relevant for all applications involving communication between two or more parts: *interoperability*. Interoperability is thus the requirement expressing the user's dream of exchanging any type of information without any technical barriers, in the simplest way. Without a standard way to perform some of the operations involved in the communication process and to structure the data exchanged, easy interoperability between terminals would be impossible. Having said that, it is clear that a standard shall specify the minimum number of tools needed to guarantee interoperability (because it is important that as many as possible non-normative technical zones exist), to allow the incorporation of technical advances, and thus to increase the lifetime of the standard, as well as to stimulate industrial technical competition. The existence of a standard also has important economic implications, because it allows the sharing of costs and investments and the acceleration of applications deployment.

MPEG has been responsible for the successful MPEG-1 (ISO/IEC 11172) and MPEG-2 (ISO/IEC 13818) standards, which have given rise to widely adopted commercial products and services, such as Video-CD, DVD, digital television, digital audio broadcasting (DAB), and MP3 (MPEG-1 Audio layer 3) players and recorders. The MPEG-4 standard (ISO/IEC 14496) is aimed at defining an audiovisual coding standard to address the emerging needs of the communication, interactive, and broadcasting service models as well as the needs of the mixed service models resulting from their convergence. The apparent convergence of the three traditionally separate application areas—communications, computing, and TV/film/entertainment—was evident in their cross-fertilization, with functionalities characteristic of each area increasingly emerging in the others (e.g., personal communications including video information or entertainment including interactive capabilities).

Following the previous successes—in fact, as a natural consequence of the vision underpinning MPEG-4—MPEG initiated in 1996 another standardization project addressing the problem of describing audiovisual content to allow the quick and efficient searching, processing, and filtering of various types of multimedia material: MPEG-7 (ISO/IEC 15938), officially called *Multimedia Content Description Interface* [N4509]. In fact, digital audiovisual information is more and more accessible to everyone, not only in terms of consumption but also in terms of production. But if it is much easier today to acquire, process, and distribute audiovisual content, it must be equally easy to access the available information, because huge amounts of audiovisual information are being generated all over the world every day. The need for a powerful way to quickly and efficiently identify, search, and filter various types of audiovisual content, by humans or machines (also using non-text-based technologies), directly follows from the urge to efficiently use the available audiovisual content and the difficulty of doing so. MPEG-7 will specify a standard way of describing various types of audiovisual information, including still pictures, video, speech, audio, graphics, 3D models, and synthetic audio, regardless of their representation format (e.g., analog or digital) and storage support (e.g.,

paper, film, or tape). In comparison with other available or emerging solutions for audiovisual content description, MPEG-7 can be mainly distinguished by (a) being general purpose, meaning its ability to describe content from many application environments; (b) its object-based representation model, meaning the capability of independently describing individual objects within a scene (be it MPEG-4 or any other format); (c) the integration of low-level and high-level features/descriptors into a single description framework, allowing it to combine the power of both types of descriptors; and (d) its extensibility, provided by the Description Definition Language, which allows MPEG-7 to keep growing, to be extended to new application areas, to answer newly emerging needs, and to integrate novel description tools [PeKo99]. The MPEG-7 standard was finalized in the summer of 2001.

Following the development of standards addressing more focused targets, MPEG acknowledged the lack of a big picture that described how the various elements building the infrastructure for the deployment of applications using multimedia content relate to each other, or even if there are missing standard specifications for some of these elements [N4333]. To address this problem, MPEG started the MPEG-21 project (first ISO/IEC 18034, now ISO/IEC 21000), formally called *Multimedia Framework*, with the aim of understanding if and how these various elements fit together, and to discuss which new standards might be required if gaps in the infrastructure exist. Once this work has been carried out, new standards will be developed for the missing elements with the involvement of other bodies, where appropriate; finally, the existing and novel standards will be integrated in the MPEG-21 multimedia framework. The MPEG-21 vision is thus to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities [N4333]. The MPEG-21 multimedia framework will identify and define the key elements needed to support the multimedia value and delivery chain, as well as the relationships between and the operations supported by them [N4511].

After briefly covering the context that motivated the birth of the MPEG-4 project, this chapter presents its major objectives in terms of functionalities, requirements, tools, and applications, as well as its organization and the sequence followed to achieve the goals defined. This chapter also addresses the MPEG modus operandi, notably its mission, principles, and specific approach to the development of standards. Finally, the objectives and working approach of the MPEG-4 Industry Forum will be presented.

1.1 MPEG-4 OBJECTIVES

Although MPEG discussions about projects beyond MPEG-2 began as early as May 1991, at the Paris MPEG meeting, it was not until September 1993 that the MPEG Applications and Operational Environments (AOE) subgroup was set up and met for the first time. The main task of this subgroup was to identify

the applications and requirements relevant to the far-term, very low bit-rate coding solution to be developed by International Organization for Standardization (ISO)/MPEG as stated in the initial MPEG-4 project description [N271]. At the same time, the near-term hybrid coding solution being developed within the International Telecommunications Union-Telecommunications Standardization Sector (ITU-T) Low Bit-rate Coding (LBC) group started producing the first results (later, the ITU-T H.263 standard [H263]). It was then generally felt that those results were close to the best performance that could be obtained by block-based, hybrid, DCT/motion-compensation video coding schemes.

In July 1994, the Grimstad MPEG meeting marked a major change in the direction of MPEG-4. Until that meeting, the main goal of MPEG-4 was to obtain a significantly better compression ratio than could be achieved by conventional coding techniques. Few people, however, believed it was possible, in the next five years, to make enough improvements over the LBC standard (H.263 and H.263+) to justify a new standard.² So the AOE subgroup was faced with the need to broaden the objectives of MPEG-4, believing that pure compression gains would not be enough to start a new MPEG standardization project. The subgroup then began an in-depth analysis of the audiovisual world trends, based on the convergence of the TV/film/entertainment, computing, and telecommunications worlds. The conclusion was that the emerging MPEG-4 coding standard should support new ways (notably content-based) of communicating, accessing, and manipulating digital audiovisual data.

1.1.1 Functionalities

Following this change of direction and the analysis made, the vision driving the MPEG-4 standard was explained through the eight new or improved functionalities described in the MPEG-4 Proposal Package Description (PPD), prepared by the time of the first MPEG-4 call for proposals in July 1995 [N998]. These eight functionalities came from an assessment of the functionalities that would be useful in future audiovisual applications, but which were not supported (or at least not well supported) by the available coding standards. The eight new or improved functionalities were clustered in three classes related to the three worlds—TV/film/entertainment, computing, and telecommunications—the convergence of which MPEG-4 wanted to address [N998, PeKo96]:

1. Content-based interactivity

- ⇒ **Content-based multimedia data access tools:** MPEG-4 shall provide efficient data access and organization based on the audiovisual content. Access tools may be indexing, hyperlinking, querying, browsing, uploading, downloading, and deleting. Sample uses include content-based

2. At least under the acceptable complexity boundaries.

retrieval of information from online libraries and travel information databases.³

- ☞ **Content-based manipulation and bitstream editing:** MPEG-4 shall provide syntax and coding schemes to support content-based manipulation and bitstream editing without the need for transcoding. This means the user should be able to select one specific object in the scene/bitstream and change some of its characteristics. Sample uses include home movie production and editing, interactive home shopping, and the insertion of a sign language interpreter or subtitles.
- ☞ **Hybrid natural and synthetic data coding:** MPEG-4 shall support efficient methods for combining synthetic scenes with natural scenes (e.g., text and graphics overlays), the ability to code and manipulate natural and synthetic audio and visual data, and decoder-controllable methods of mixing synthetic data with ordinary video and audio (allowing for interactivity). For example, in virtual reality applications, animations and synthetic audio (e.g., MIDI) can be mixed with ordinary audio and video in games, and graphics can be rendered from different viewpoints.
- ☞ **Improved temporal random access:** MPEG-4 shall provide efficient methods to randomly access, within a limited time and with fine resolution, parts (e.g., frames or objects) from an audiovisual sequence. Example usage: audiovisual data can be randomly accessed from a remote terminal over limited-capacity media, a fast-forward can be performed on a single audiovisual object in the sequence.

2. Compression efficiency

- ☞ **Improved coding efficiency:** The growth of mobile networks creates an ongoing demand for improved coding efficiency; therefore, MPEG-4 set as its target providing subjectively better audiovisual quality compared to existing or other emerging standards (such as H.263), at comparable bit rates. Sample uses include efficient transmission of audiovisual data on low-bandwidth channels and efficient storage of audiovisual data on limited-capacity media, such as chip cards.
- ☞ **Coding of multiple concurrent data streams:** MPEG-4 shall provide the ability to efficiently code multiple views/soundtracks of a scene as well as sufficient synchronization between the resulting elementary streams.⁴ For stereoscopic and multiview video applications, MPEG-4 shall include the ability to exploit redundancy in multiple views of the

3. Although this is one of the eight initial MPEG-4 functionalities, it relates more to the tools developed later in the context of the MPEG-7 standard [N4509].

4. This seems to be the only initial MPEG-4 functionality that has not been addressed at the end of 2001, as the companies participating in MPEG have shown no significant interest in it.

same scene, also permitting solutions that allow compatibility with normal video. Sample uses include multimedia entertainment (e.g., virtual reality games and 3D movies), training and flight simulations, multimedia presentations, and education.

3. Universal access

- ☛ **Robustness in error-prone environments:** Because universal accessibility implies access to applications over many wireless and wired networks and storage media, MPEG-4 shall provide an error robustness capability. Particularly, sufficient error robustness shall be provided for low bit-rate applications under severe error conditions. Sample uses include transmission from a database over a wireless network, communicating with a mobile terminal, and gathering audiovisual data from a remote location.
- ☛ **Content-based scalability:** MPEG-4 shall provide the ability to achieve scalability with a fine granularity in content, spatial resolution, temporal resolution, quality, and complexity. Content-scalability may imply the existence of a prioritization of the objects in the scene. Sample uses include user selection of decoded quality of individual objects in the scene and database browsing at different scales, resolutions, and qualities.

These functionalities were essential to shaping the MPEG-4 vision, balancing completely new functionalities with more traditional ones, and thus allowing a bridge from the past to the future not only in terms of functionalities but also in terms of tools and experts.

1.1.2 Requirements

Following the identification of the fundamental MPEG-4 functionalities, MPEG started a requirements development process, which has been continuously evolving since then. The requirements serve to drive the tools development process and assure that the right technology is being specified: Tools will be developed to fulfill the identified requirements, and no tools that do not address any requirement will be defined.

By the middle of 2001, the MPEG-4 requirements were structured as shown in Table 1.1. The requirements are organized in terms of major technical areas, which do not directly correspond either to Parts of the MPEG-4 standard or to MPEG working subgroups. Table 1.1 gives only a general flavor of the requirements (for the details, see [N4319]).

Within each category, the requirements are to be fulfilled by the set of tools selected for the standard, and not all requirements must be addressed by each individual tool. It is the right combination of tools that allows building the algorithms that can address the specific needs of a certain class of applications.

Table 1.1 MPEG-4 requirements [N4319]

Requirements for systems	
Flexibility	Multipoint operation
Multiplexing of audio, visual, and other information	Object content information (OCI)
Composition of audio and visual objects	Video-related metadata
Application texture	Delay
Downloading	Configuration modes
User interaction	Priority of audiovisual objects
Media interworking	Dynamic resource management
Compatibility	Reference to associated MPEG-7 data
Robustness to information errors and losses	File format
Object-based bitstream manipulation and editing	Textual format
Requirements for natural video objects	
Object-based representation	Object-based coding flexibility
Video content	Object-based scalability
Object-based bitstream manipulation and editing	Delay modes
Object-based random access	Formats
Object quality and fidelity	Bit-rate modes
Coding of multiple concurrent data streams	Complexity modes
Robustness to information errors and losses	Still images
	Tandem coding
Requirements for synthetic video objects	
Types of synthetic video objects	Text overlay
2D/3D mesh compression	Image and graphics overlay
Definition and animation parameter compression	View-dependent texture scalability
Texture mapping	Geometrical transformations
	Video object tracking
Requirements for natural audio objects	
Object-based representation	Robustness to information errors and losses
Audio content	Delay modes
Object-based bitstream editing and manipulation	Complexity modes
Object-based scalability	Bit-rate modes
Object-based random access and user controls	Downmix [†]
Time scale change	Transcoding
Pitch change	Tandem coding
	Audio formats
	Improved coding efficiency

Table 1.1 MPEG-4 requirements [N4319] (Continued)

Requirements for synthetic audio objects	
Low bit-rate speech	Text to speech
Synthetic speech data	Sound synthesis
Requirements for delivery multimedia integration format (DMIF)	
Connectivity	End-to-end Quality of Service (QoS) management
Transparency	
Application service enablement	Network-based stream processing and management
Requirements for MPEG-J	
Functional requirements	Byte code execution
Byte code delivery	Event mechanism
Authentication	
Requirements for multiuser environments	
Scene graph representation	IPMP and sharing tools
Audiovisual objects and avatars representation	Application programming interfaces
Delivery and stream management	
Requirements for animation framework	
Enhanced texture mapping	Reusability of scene graph nodes and animation streams
Animation support	Persistence
High-level shape representation	Compression of animated objects
Requirements for intellectual property management and protection (IPMP)	
Identification of intellectual property	Intellectual property management and protection interfaces
Intellectual property management and protection hooks	

^aRelates to the repeated encoding and decoding (cascading) of audio or video material, either with the same or different codecs.

^bRelates to the ability to reduce the number of channels to a configuration with a lower number of channels for presentation purposes (e.g., listening to multichannel audio using stereophonic reproduction).