UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE, LLC
Petitioner

v.

WITHROW NETWORKS, INC.
Patent Owner

U.S. Patent No. 10,771,849 B2
Issue Date: September 8, 2020

Title: MULTIMEDIA SYSTEM FOR MOBILE CLIENT PLATFORMS

DECLARATION OF INGRID HSIEH-YEE, PH.D.

TABLE OF CONTENTS

		Page
I.	INTRODUCTION AND QUALIFICATIONS	6
	A. Qualifications and Professional Experience	6
II.	BACKGROUND OF LIBRARY CATALOGING	8
	A. Machine-Readable Cataloging ("MARC")	8
	B. Library Online Catalogs	10
	C. Scope of this Declaration	15
	D. Evidence Considered in Forming My Opinions	19
III.	AUTHENTICITY AND PUBLIC AVAILABILITY OF *MATTAVELLI* (EX. 1004)	20
	A. Authentication	20
	B. Records of Library of Congress	22
	C. Summary of My Opinion on *Mattavelli*	27
IV.	AUTHENTICITY AND PUBLIC AVAILABILITY OF *WANG* (EX. 1007)	27
	A. Authentication	27
	B. IEEE Metadata Record	30
	C. Copyright Registration Record	33
	D. Records of Linda Hall Library	34
	E. Summary of My Opinion on *Wang*	38
V.	AUTHENTICITY AND PUBLIC AVAILABILITY OF *WOLTERS* (EX. 1041)	39
	A. Authentication	39
	B. Records of Linda Hall Library	41
	C. Records of the Canadian National Science Library	45
	D. Internet Archive Records	49
	E. Usage Records	53
	F. Summary of My Opinion on *Wolters*	53
VI.	PUBLIC AVAILABILITY OF *WATKINSON* (EX. 1005)	55
	A. Library of Congress Records	55

Google Exhibit No. 1063
Page 0002

B. Summary of My Opinion on *Watkinson* 59
VII. PUBLIC AVAILABILITY OF *GOURLEY* (EX. 1006) 59
A. Library of Congress Records.....60
B. Copyright Registration Record.....63
C. Summary of My Opinion on *Gourley* 63
VIII. PUBLIC AVAILABILITY OF *MPEG GUIDE* (EX. 1014).....64
A. Internet Archive Records 65
B. Summary of My Opinion on *MPEG Guide* 67
IX. PUBLIC AVAILABILITY OF *MIETENS* (EX. 1018).....68
A. Record from American University.....68
B. Summary of My Opinion on *Mietens*.....70
X. PUBLIC AVAILABILITY OF *HUI* (EX. 1019) 70
A. Date Stamped Copy from Library of Congress 71
B. Library of Congress Records.....72
C. Usage Records.....74
D. Summary of My Opinion on *Hui* 75
XI. PUBLIC AVAILABILITY OF *SYMES* (EX. 1020) 75
A. Library of Congress Records.....76
B. Summary of My Opinion on *Symes* 79
XII. PUBLIC AVAILABILITY OF *PEREIRA* (EX. 1021) 80
A. Library of Congress Records.....80
B. Copyright Registration Record.....83
C. Summary of My Opinion on *Pereira*.....84
XIII. PUBLIC AVAILABILITY OF *HUNT* (EX. 1022).....84
A. Library of Congress Records.....85
B. Copyright Registration Record.....88
C. Summary of My Opinion on *Hunt* 89
XIV. PUBLIC AVAILABILITY OF *COMER* (EX. 1023) 89
A. Library of Congress Records.....90

Google Exhibit No. 1063
Page 0003

B. Copyright Registration Record....93
C. Summary of My Opinion on *Comer* 93
XV. PUBLIC AVAILABILITY OF *STOCKHAMMER* (EX. 1027)....94
A. Date Stamped Copy from Library of Congress 94
B. Library of Congress Records....96
C. Summary of My Opinion on *Stockhammer* 98
XVI. PUBLIC AVAILABILITY OF *RICHARDSON* (EX. 1028) 98
A. Library Records of University of Essex....99
B. Library Records of the Texas A&M University 101
C. Summary of My Opinion on *Richardson*....104
XVII. PUBLIC AVAILABILITY OF *BOURGE* (EX. 1029) 104
A. Linda Hall Library Copy 105
B. Linda Hall Library Records....106
C. Summary of My Opinion on *Bourge* 108
XVIII. PUBLIC AVAILABILITY OF *CEN* (EX. 1032)....109
A. Linda Hall Library Copy 109
B. Linda Hall Library Records....111
C. Summary of My Opinion on Cen....113
XIX. PUBLIC AVAILABILITY OF *LU* (EX. 1033) 114
A. Linda Hall Library Copy 114
B. Linda Hall Library Records....116
C. Summary of My Opinion on *Lu* 118
XX. PUBLIC AVAILABILITY OF *WETZEL* (EX. 1038) 118
A. Linda Hall Library Copy 119
B. Linda Hall Library Records....120
C. Copyright Registration Record....124
D. Summary of My Opinion on *Wetzel*....124
XXI. PUBLIC AVAILABILITY OF *PAIK* (EX. 1039) 125
A. Date Stamped Copy from Library of Congress 126

Google Exhibit No. 1063
Page 0004

B. Library of Congress Records....127
C. Copyright Registration Record....129
D. Summary of My Opinion on *Paik*....130
XXII. PUBLIC AVAILABILITY OF *LEE* (EX. 1040)....130
A. Date Stamped Copy from Library of Congress....131
B. Library of Congress Records....132
C. Summary of My Opinion on *Lee*....134
XXIII. PUBLIC AVAILABILITY OF *DE SMET 2004* (EX. 1042)....135
A. Records of British Library....136
B. Summary of My Opinion on *De Smet 2004*....137
XXIV. PUBLIC AVAILABILITY OF *DE SMET* (EX. 1043)....138
A. Date Stamped Copy from Library of Congress....138
B. Library of Congress Records....140
C. Copyright Registration Record....142
D. Summary of My Opinion on *De Smet*....142
XXV. PUBLIC AVAILABILITY OF *MPEG OVERVIEW* (EX. 1044)....143
A. Internet Archive Records....144
B. Summary of My Opinion on *MPEG Overview*....146
XXVI. PUBLIC AVAILABILITY OF *MESAROVIC* (EX. 1047)....146
A. Records of German National Library of Science and Technology....146
B. Summary of My Opinion on *Mesarovic*....149
XXVII. CONCLUSION....150

Google Exhibit No. 1063
Page 0005

I, Ingrid Hsieh-Yee, declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

1. I am over the age of eighteen and otherwise competent to make this declaration.

2. I have been retained by counsel for Petitioner to provide my expert opinion in connection with the above-captioned *inter partes* review ("IPR") proceeding regarding the public availability of a certain prior art references in relation to U.S. Patent No. 10,771,849 (the "'849 patent").

3. I am being compensated for my time in connection with this IPR at my current consulting rate. I am also being reimbursed for any reasonable expenses associated with my work and testimony in this investigation. My compensation is not contingent on the results of my study, the substance of my opinions, or the outcome of this matter.

A. Qualifications and Professional Experience

4. My complete qualifications and professional experience are described in my academic curriculum vitae (**Appendix A**). The following is a brief summary of my relevant qualifications and professional experience.

5. I was a professor in the Department of Library and Information Science at the Catholic University of America for 32 years and was conferred the title of Professor Emerita in December 2022. I have experience working in an academic

library, a medical library, and a legislative library. I hold a Ph.D. and a Masters in Library and Information Studies from the University of Wisconsin-Madison.

6. I am an expert on library cataloging and classification and have published two editions on this subject, *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide* (2000 and 2006). I taught a variety of courses, including: Cataloging and Classification, Advanced Cataloging and Classification, Organization of Internet Resources, Organization of Information, Metadata, Digital Content Creation and Management, Internet Searches and Web Design, Information Literacy Instruction, Advanced Information Retrieval and Analysis Strategies, and The Information Professions in Society. I am familiar with metadata schema design and implementation. In my teaching, I covered the design and implementation of metadata in databases, search engines, digital repositories, digital libraries, and digital archives. I also covered how information organization affects the discovery and access to digital resources on the Internet. My research interests cover cataloging and classification, information organization, metadata, information retrieval, information architecture, digital collections, scholarly communication, social media, user interaction with information systems, and other related topics.

Google Exhibit No. 1063
Page 0007

II. BACKGROUND OF LIBRARY CATALOGING

A. Machine-Readable Cataloging ("MARC")

7. I am very familiar with a library cataloging encoding standard known as the "Machine-Readable Cataloging" standard, also known as "MARC," which became the national standard for sharing bibliographic data in the United States by 1971 and the international standard by 1973. MARC is the primary communications protocol for the transfer and storage of bibliographic metadata in libraries. Experts in my field reasonably rely upon MARC records when forming their opinions.

8. A MARC record consists of several fields, each of which contains specific data about the work. Each field is identified by a standardized, unique, three-digit code corresponding to the type of data that follows. **Appendix B** is a true and correct copy of Parts VII to X of "Understanding MARC Bibliographic: Machine-Readable Cataloging" (https://www.loc.gov/marc/umb/um07to10.html), a brief description and tutorial published by the Library of Congress in 2009 that explains commonly used MARC fields. For example, the personal author of a work is recorded in Field 100, the title is recorded in Field 245, publisher information is recorded in Field 260, the physical volume and characteristics of a publication are recorded in Field 300, and topical subjects are recorded in Fields 650. More details of MARC21 format for Bibliographic Data are available on the Library of Congress website at https://www.loc.gov/marc/bibliographic/.

9. I am very familiar with library cataloging and processing practices, the relationships between OCLC (Online Computer Library Center) and library catalogs, and how libraries create and share their records to help the public discover and access information.

10. OCLC is the largest bibliographic network of the world, and its WorldCat database contains more than 559 million records from thousands of member institutions (many of which are libraries of some type) in more than 100 countries. OCLC was founded in 1967 to promote and support library cooperation. According to the third article of the "Amended Articles of Incorporation of OCLC [Online Computer Library Center], Inc.," OCLC was created "to establish, maintain, and operate a computerized library network and to promote the evolution of library use, of libraries themselves, and of librarianship, and to provide processes and products for the benefit of library users and libraries, including such objectives as increasing availability of library resources to individual library patrons and reducing the rate of rise of library per-unit costs, all for the fundamental public purpose of furthering ease of access to and use of the ever-expanding body of worldwide scientific, literary, and educational knowledge and information" (source: https://www.oclc.org/content/dam/oclc/membership/articles-of-incorporation.pdf).

11. OCLC members can contribute original cataloging records in MARC to the system or derive cataloging records from existing records, an activity referred

to as "copy cataloging." When an OCLC participating institution acquires a work, it can create an original MARC record for the work in OCLC's Connexion system (a system for catalogers to create and share MARC records), and the system will automatically generate a code for the date of record creation in the yymmdd format, and the creating library's OCLC symbol is recorded in subfield "a" of the 040 field. Once the MARC record is in Connexion, it becomes available to other OCLC members for adoption to their local online catalogs (*i.e.*, copy cataloging).

12. After a MARC record is created in Connexion, it also becomes searchable and viewable on WorldCat.org, which is a free web portal for users to explore more than 10,000 libraries worldwide. Records in WorldCat.org, however, are not presented in MARC fields. Instead, the data elements are labeled to help users interpret the records. Thus, the information stored in MARC records in Connexion is available to the interested public through the user-friendly WorldCat web portal. WorldCat records can also be searched on OCLC FirstSearch, a subscription-based search system that provides access to many proprietary databases, including the WorldCat database.

B. Library Online Catalogs

13. Library online catalogs gained acceptance in the early 1980s, and many libraries migrated their systems to the World Wide Web in the mid-1990s. Library online catalogs are based on MARC records that represent their collections to help

Google Exhibit No. 1063
Page 0010

the public understand what materials are publicly accessible in those libraries. Most libraries with online catalogs have made their catalogs freely available on the Web. These online catalogs offer user-friendly search interfaces. Strong user interest in keyword searches and the popularity of Google have led to the "googlization" of library search systems. As a result, many library catalogs nowadays provide a single search box for users to conduct keyword searches, with additional support for searches by author, title, subject terms, and other data elements such as ISBN (International Standard Book Number). Library catalogs these days also offer features for users to narrow their search results by language, year, format, and other elements. Many libraries display MARC records on their online catalogs with labels for the data elements to help the public interpret MARC records. Many libraries also offer the option to display MARC records in MARC fields.

14. Libraries create MARC records for works they acquire, including books, serials, motion pictures, and publications in other formats. Monograph cataloging is fairly common in libraries, and most libraries make a newly cataloged monograph available to the public soon after the cataloging work is completed, usually within a week. If processing delays occur, the newly cataloged monograph would be publicly accessible no later than one month after the cataloging work is completed. Libraries can create original cataloging records or use an existing record in the OCLC WorldCat database to create a copy cataloging record. As soon as the

Google Exhibit No. 1063
Page 0011

cataloging record is completed, it is added to the library's online catalog for users. If the record is an original record, it is also entered into the OCLC system. If it is a copy cataloging record, the library's holding symbol is attached to the existing original record in the OCLC system to facilitate searching and interlibrary loan.

15. The cataloging of serials and the serial check-in process are discussed here to show how libraries usually provide access to newly received serial issues. According to the glossary of the *RDA: Resource Description and Access* cataloging standard, a serial is "a mode of issuance of a manifestation issued in successive parts, usually bearing numbering, that has no predetermined conclusion. A serial includes a periodical, monographic series, newspaper, etc." (RDA Glossary at https://original.rdatoolkit.org/rdagloss_rdaregistry.info-termList-ModeIssue-1003_1.html (subscription required for access); cited in https://www.ccslib.org/Catalogers/index.php/Category:Serials). Because the publisher of a serial makes new issues of the serial available successively, a customary cataloging practice is to create one bibliographic record for the serial, and the serial record (encoded in MARC) typically provides information on the beginning date and frequency of the serial, not the dates of individual issues. In other words, libraries typically do not create MARC records for individual issues of a serial. Instead, they rely on a serial check-in system to track the receipt of new issues. A common check-in practice is to date stamp a new issue when it arrives. This practice

Google Exhibit No. 1063
Page 0012

has become automated since the late 1990s, and libraries now vary in how they share the receipt date of a new serial issue with the public. Some libraries use a date stamp, some affix a label to indicate the receipt date, some pencil in the receipt date, and some do not provide the information to the public.

16. The serial check-in process usually takes less than an hour, and one of the steps involves placing a date stamp on the new issue to document the date the issue is checked in. After that, the holdings information of the serial is updated in the library's catalog so that users know which issues are available for request or access. After serial check-in is completed, the new issue is placed on the shelf with the previous issues of the serial. Libraries with a public periodical room typically place new issues in the periodical room for easy user access. Because information presented in serials often reflects latest discovery, a general practice of libraries is to make new issues of serials available for user access soon after they are checked in, usually within a week.

17. I am personally familiar with many online catalogs, databases, and search engines. In preparing for this declaration, I used the following authoritative information systems to search for records:

- Google Scholar (https://scholar.google.com/)
- IEEE Xplore Digital Library (https://ieeexplore.ieee.org/Xplore/home.jsp)

Google Exhibit No. 1063
Page 0013

- Internet Archive (https://archive.org/)
- Libraries.org, a directory of libraries throughout the world (https://librarytechnology.org/libraries/)
- OCLC FirstSearch (subscription required for access; https://firstsearch.oclc.org/)
- Online catalog of Technische Informationsbibliothek (TIB) (https://opac.tib.eu/DB=1/LNG=DU/)
- Online catalog of the Federal Science Libraries Network of Canada (https://science-libraries.canada.ca/eng/home/)
- Online catalog of the Library of Congress (https://catalog.loc.gov/)
- Online catalog of the Linda Hall Library (https://catalog.lindahall.org/discovery/search?vid=01LINDAHALL_INST:LHL)
- Online catalog of the Texas A&M University Libraries (https://catalog.library.tamu.edu/)
- Online catalog of the University of Essex Library (https://essex.primo.exlibrisgroup.com/discovery/search?vid=44UOES_INST:UOES&lang=en)
- Public catalog of the US Copyright Office (https://cocatalog.loc.gov/)
- WorldCat.org (https://search.worldcat.org/)

Google Exhibit No. 1063
Page 0014

C. Scope of this Declaration

18. I have been asked to offer an opinion on the public availability dates of the following documents:

Exhibit	Document
1004	Mattavelli, M., & Brünettem, S. (1998, September). Computational graceful degradation for video decoding. In *Signal Processing IX: Theories and Applications: Proceedings of EUSIPCO-1998*, pp. 1-4 ("*Mattavelli*")
1007	Wang, S. H., Peng, W. H., He, Y., Lin, G. Y., Lin, C. Y., Chang, S. C., ... & Chiang, T. (2003, December). A platform-based MPEG-4 advanced video coding (AVC) decoder with block level pipelining. In *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*, Vol. 1, pp. 51-55 ("*Wang*")
1041	Wolters, M., Kjorling, K., Homm, D., & Purnhagen, H. (2003, October). A closer look into MPEG-4 High Efficiency AAC. In *Audio Engineering Society Convention 115*, Convention Paper 5871 ("*Wolters*")
1005	Watkinson, J. (2001). *The MPEG Handbook: MPEG-1, MPEG-2, MPEG-4.* Focal Press. ("*Watkinson*")
1006	Gourley, D., & Totty, B. (2002). *HTTP: The Definitive Guide*. First Edition. O'Reilly. ("*Gourley*")
1014	A Guide to MPEG Fundamentals and Protocol Analysis (including DVB and ATSC) (2000) ("*MPEG Guide*")
1018	Mietens, S. O., With, de, P. H. N., & Hentschel, C. (2004, March). New complexity scalable MPEG encoding techniques for mobile applications, *EURASIP Journal on Applied Signal Processing*, 2004(2), 236-252 ("*Mietens*")

Google Exhibit No. 1063
Page 0015

Exhibit	Document
1019	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. ("*Hui*")
1020	Symes, P. (2004). Digital Video Compression. McGraw-Hill. ("*Symes*")
1021	Pereira, F. C., & Ebrahimi, T. (2002). *The MPEG-4 Book*. Prentice Hall PTR. ("*Pereira*")
1022	Hunt, C., & Thompson, R. B. (1998). *Windows NT TCP/IP Network Administration*. First Edition. O'Reilly. ("*Hunt*")
1023	Comer, D. E. (2000). *Internetworking with TCP/IP. Volume 1: Principles, Protocols, and Architecture*. Fourth edition. Prentice Hall. ("*Comer*")
1027	Stockhammer, T., Hannuksela, M. M., & Wiegand, T. (2003, July). H. 264/AVC in wireless environments. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(7), 657-673("*Stockhammer*")
1028	Richardson, I. E. G. (2003). H.264 and MPEG-4 Video Compression. Wiley. ("*Richardson*")
1029	Bourge, A., & Jung, J. (2004, January). Low-power H. 264 video decoder with graceful degradation. In *Visual Communications and Image Processing 2004*, Proceedings of SPIE Vol. 5308, pp. 372-383 ("*Bourge*")
1032	Cen, S., Walpole, J., & Pu, C. (1997, December). Flow and congestion control for internet media streaming applications. In *Multimedia Computing and Networking 1998*, Proceedings of SPIE Vol. 3310, pp. 250-264 ("*Cen*").

Google Exhibit No. 1063
Page 0016

Exhibit	Document
1033	Lu, J. (2000, April). Signal processing for Internet video streaming: A review. *Image and Video Communications and Processing 2000*, Proceedings of SPIE Vol. 3974, pp. 246-25 ("*Lu*").
1038	Wetzel, A. T., & Schell, M. R. (1996). Consumer applications of the IEEE 1394 Serial Bus, and A 1394/DV Video editing system ("*Wetzel*")
1039	Paik, W. K., & Hwang, S. Y. (1999, November). Design of a novel synthesis filter for real-time MPEG-2 audio decoder implementation on a DSP chip, *IEEE Transactions on Consumer Electronics*, *45*(4), 1119-1129 ("*Paik*").
1040	Lee, K. S., Park, Y. C., & Youn, D. H. (2002, August). Software optimization of the MPEG-audio decoder using a 32-bit MCU RISC processor, *IEEE Transactions on Consumer Electronics*, *48*(3), 671-676 ("*Lee*").
1042	De Smet, P., Rooms, F., Luong, H. Q., & Philips, W. (2004, October). Do not zero-pute: An efficient homespun MPEG-Audio layer II decoding and optimization strategy. In *Proceedings of the 12th annual ACM international conference on* Multimedia, pp. 376-379 ("*De Smet2004*").
1043	De Smet, P, & Bruyland, I. (2003, October). Optimized recursive subband synthesis windowing for implementing efficient MPEG audio decoders. *IEEE Signal Processing Letters*, vol. 10, no. 10, 303-306 ("*De Smet2003*").
1044	Overview of the MPEG-4 standard. Robert Koenen (editor). ISO/IEC JTC1/SC29/WG11 N4668. March 2002. Version 21 ("*MPEG Overview*")
1047	Mesarovic, V., Rao, R., Dokic, M., & Deo, S. (2001, April). Optimized Huffman Decoding for AAC. In *Audio Engineering Society Conference: UK 16th Conference: Silicon for Audio*. Audio Engineering Society. pp. 117-122 ("*Mesarovic*").

Google Exhibit No. 1063
Page 0017

19. I am informed by counsel and understand that a given reference is "published" upon a satisfactory showing that such a document has been disseminated or otherwise made publicly available to the extent that persons interested and ordinarily skilled in the subject matter or art could locate it exercising reasonable diligence and obtain the document. I have also been informed by counsel and understand that materials available in a library constitute "printed publications" if they are catalogued and indexed according to general library practices and protocols that make the references available and accessible to members of the interested public.

20. I have been informed by counsel that the '849 patent was filed in the United States and that it claims April 18, 2005 as its earliest priority date. I have assessed public accessibility of the documents referenced below as of this claimed priority date.

21. I have been informed by counsel that the subject matter of the '849 patent and this proceeding generally relates to multimedia streaming and related technologies, including network communications and data compression methods.

22. I have been informed by counsel that a "person of ordinary skill in the art at the time of the invention" (POSA) is a hypothetical person who is presumed to be familiar with the relevant field and its literature at the time of the inventions. I have been informed by counsel that this hypothetical person is also a person of

Google Exhibit No. 1063
Page 0018

ordinary creativity, capable of understanding the scientific principles and literature applicable to the pertinent field.

23. I have been informed by counsel that POSA as of April 18, 2005 would have knowledge of the scientific literature and have skills relating to multimedia streaming and related technologies. I have been informed and understand that a person of ordinary skill in this subject matter or art would have had a bachelor's degree in computer science, computer engineering, or electrical engineering, or the equivalent, and at least two years of work experience in networking or streaming media systems, particularly audio and video, over the Internet. Additional education could have substituted for professional experience, and significant work experience could have substituted for formal education.

24. It is my opinion that such a person would have been actively engaged in learning about the field, possibly through formal instruction and through bibliographic resources. By at least April 18, 2005, such a person would have had access to a vast array of print and electronic resources, including at least the documents referenced below.

D. Evidence Considered in Forming My Opinions

25. In the preparation of this declaration, I have reviewed the exhibits references in paragraph 18 above, and any other documents I reference herein. Each of these is a type of material that experts in my field would reasonably rely upon

Google Exhibit No. 1063
Page 0019

when forming their opinions. I provided a further description of materials I considered in **Appendix C**. As shown in Appendix C, I have labeled and organized the additional materials I relied on associated with a particular exhibit (*e.g.*, Ex. 1004) to include a trailing letter on the respective exhibit number (*e.g.*, Ex. 1004A, 1004B, etc.). These additional materials that I have relied on have been included with my declaration.

III. Authenticity and Public Availability of *Mattavelli* (Ex. 1004)

A. Authentication

26. **Ex. 1004** is a true and correct copy of "Computational Graceful Degradation for Video Decoding" by M. Mattavelli & S. Brunetton, in *Signal Processing IX: Theories and Applications: Proceedings of EUSIPCO-1998*, pp. 2309-2312, that I received from counsel. This copy is presented as **Ex. 1004** in this declaration.

27. Page 1 is internal page 2309 that shows the title, authors, and abstract of *Mattavelli*. **Ex. 1004** shows this article has four pages, including three figures, five equations, and six references.

28. Page 5 of **Ex. 1004** shows the spine of a bound volume identified as Signal Processing IX, European Signal Processing Conference, that has a call number of "TK5102.9.E97 1998," and this volume is volume 4. Page 6 shows the cover of this conference proceedings, page 8 is the title page that shows "Signal Processing

Google Exhibit No. 1063
Page 0020

IX" as the main title, "Theories and Applications" as the subtitle, and this volume is "Volume IV of IV" of the Proceedings of Eusipco-98, Ninth European Signal Processing Conference that took place in Rhodes, Greece, from 8 to 11 of September 1998. The title page shows four editors edited the proceedings, three organizations that sponsored the conference, and that Typorama Editions published this proceeding in 1998. The left side of page 8 is the copyright page that shows Eurasip and Typorama Editions hold the copyright for this work. It also shows the ISBN for the four-volume set and the ISBN for each volume. The left side of page 9 is a page that shows a hand-written call number for this volume and a date stamp of "Library of Congress DEC 16. 1998 GIFT" that indicates the Library of Congress received this volume on December 16, 1998. The full table of contents begins on the right side of page 15 (internal page xix) and ends on the left side of page 37 (internal page lvii). The left side of page 35 (internal page lviii) shows that "Computational Graceful Degradation for Video Decoding" by Mattavelli and Brunetton is part of FA S-3, Image and Video Coding, and begins on internal page 2309.

29. To authenticate **Ex. 1004**, I searched for *Mattavelli* in Google Scholar and the record shows EURASIP (European Association for Signal Processing) offers a free copy of this article at https://eurasip.org/Proceedings/Eusipco/Eusipco1998/sessions/F%20A/FA%20S-3/2309/694.PDF. This EURASIP copy of *Mattavelli* is presented as **Ex. 1004A**.

Google Exhibit No. 1063
Page 0021

30. **Ex. 1004A** is a true and correct copy of *Mattavelli* and the pages are not numbered. Page 1 shows the title, authors, and abstracts. This exhibit shows *Mattavelli* has four pages, including three figures, five equations, and six references. I have closely compared **Ex. 1004** and **Ex. 1004A** and found them to contain the same content for *Mattavelli*.

31. **Ex. 1004** is in a condition that creates no suspicion about its authenticity. Specifically, this copy is not missing any intermediate pages, and there are no visible alterations to the document. Furthermore, the copy of *Mattavelli* (**Ex. 1004**) has the same content as **Ex. 1004A**, which is provided by the European Association for Signal Processing. Based on my review and analysis, it is my opinion that **Ex. 1004** is authentic.

B. Records of Library of Congress

32. To determine the public availability date of *Mattavelli*, I obtained records for the conference proceedings that contain *Mattavelli* from the online catalog of the Library of Congress. **Ex. 1004B** is a true and correct copy of the serial bibliographic and MARC records for *Signal Processing IX*, which is the proceedings of EUSIPCO-98. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

33. Pages 1 and 2 of **Ex. 1004B** are the bibliographic record that shows the

Google Exhibit No. 1063
Page 0022

Library of Congress cataloged this volume of proceedings as a book. The record shows the title of this work is "Signal processing IX: theories and applications: proceedings of EUSIPCO-98, Ninth European Signal Processing Conference, Rhodes, Greece, 8-11 September 1998," and that Typorama Editions of Patras, Greece, published this work in 1998. The record also shows this work has four physical volumes and a companion CD-ROM, and the ISBN (International Standard Book Number) for the set is "9607620054." The Item Availability area on page 2 shows the Library of Congress has one copy of this proceeding, the call number is "TK5102.9 .E97 1998," users can request access at the Jefferson or Adams Building Reading Rooms, and the status of the copy is "Not Charged" (meaning available for user request).

34. Pages 3 and 4 are the MARC record for the conference proceedings containing *Mattavelli*. The first six digits of Field 008 show the MARC record was created on "981216" (*i.e.*, December 16, 1998). Field 005 is a machine-generated field that records the last time this library modified this MARC record, and it shows the Library of Congress last modified this MARC record on "19990122143936" (*i.e.*, January 22, 1999, at 2 p.m., 39 minutes, and 36 seconds). Field 955 is a field for local cataloging notes, and it shows the record was sent to "ddc" (the Dewey Decimal Classification unit) on "01-08-99" (*i.e.*, January 8, 1999). Subfields "a" and "c" of Field 040 show "DLC" created the original record. According to the *Directory of*

Google Exhibit No. 1063
Page 0023

OCLC Members (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library symbol for the Library of Congress. The date stamp in **Ex. 1004** shows the library received a physical copy of the conference proceedings on "DEC 16. 1998." This date stamp and the data from Fields 008, 005, 955, and 040 inform my opinion that the Library of Congress created an initial MARC record for the conference proceedings containing *Mattavelli* on December 16, 1998, received their copy of the proceedings on December 16, 1998, and last modified their record on January 22, 1999. A customary library practice is to process a newly cataloged work promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that *Signal Processing IX* that contains *Mattavelli* became discoverable by January 22, 1999, in the online catalog of the Library of Congress, and the physical copy, after processing, would become publicly accessible no later than April 22, 1999.

35. Field 020 shows the ISBN of *Signal Processing IX*. Field 111 shows the conference of EUSIPCO-98 is recorded as the corporate body for this work. Field 245 shows the title and subtitles, and Fields 246 show two variant titles are included to help users discover this conference proceedings. Field 260 shows the publisher information, Field 300 shows the physical details, Field 504 shows the work includes

Google Exhibit No. 1063
Page 0024

bibliographical references and indexes, and Field 700 shows the first editor is included to help users discover this work.

36. The subjects of this conference proceedings are represented in Field 050 by a Library of Congress Classification (LCC) number of "TK5102.9" that represents periodicals on "Signal processing" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of "621.382/2" that represents "Communications engineering" (source: Dewey Decimal Classification 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf). Subjects are also represented in Field 650 by one Library of Congress subject heading with "Signal processing" as the main heading, followed by "Congresses" as a form subheading encoded in subfield "x" ("|x") to indicate the main topic is treated at a conference. These subject representations enable users to conduct subject searches to discover the conference proceedings that contain *Mattavelli*.

37. This MARC record (**Ex. 1004B**) shows that the Library of Congress modified their MARC record on January 22, 1999, meaning *Signal Processing IX* became discoverable by January 22, 1999, and users could discover it by the ISBN, conference name, title, variant titles, and the first editor. They could also use the LCC number, DDC number, and Library of Congress subject heading to conduct subject

Google Exhibit No. 1063
Page 0025

searches to discover this work.

38. The date stamp on **Ex. 1004** shows the Library of Congress received this conference proceedings on December 16, 1998, and the MARC record shows the Library of Congress last modified their record on January 22, 1999. It is therefore my opinion that the conference proceedings containing *Mattavelli* was publicly accessible at the Library of Congress by April 22, 1999.

39. As evidence that the Library of Congress had an online catalog in operation in 1999 when the library last modified its record for the conference proceedings containing *Mattavelli*, I obtained a profile of the library at https://librarytechnology.org/library/2263 from libraries.org, a directory of libraries throughout the world. I personally located, identified, and obtained this profile. This is the type of record experts in my field would reasonably rely upon when forming their opinions. This profile is presented as **Ex. 1004C** in this declaration.

40. The Technology Profile on page 2 of **Ex. 1004C** shows the Library of Congress began using Voyager as their online catalog in 1998, and changed to Folio – EBSCO Information Services in 2023. Based on personal knowledge and experience, I can confirm that the Voyager system was in operation in 1999 at the Library of Congress. These dates mean that in 1999 interested users were able to use the Voyager system to search for the conference proceedings containing *Mattavelli* and access this article at the Library of Congress.

Google Exhibit No. 1063
Page 0026

C. Summary of My Opinion on *Mattavelli*

41. The copy of *Mattavelli* obtained from the Library of Congress (**Ex. 1004**) is authentic because it contains no signs to raise suspicion about its authenticity. In addition, this copy has the same content as the copy that is freely available from the European Association for Signal Processing website (**Ex. 1004A**).

42. The MARC record of the Library of Congress (**Ex. 1004B**) shows the library received the conference proceedings containing *Mattavelli* on December 16, 1998, and the MARC record was last modified on January 22, 1999. These data inform my opinion that the conference proceedings containing *Mattavelli* became discoverable in the online catalog of the Library of Congress by January 22, 1999, and the physical copy of the conference proceedings, after processing, became publicly accessible no later than April 22, 1999.

IV. Authenticity and Public Availability of *Wang* (Ex. 1007)

A. Authentication

43. **Ex. 1007** is a true and correct copy of "A Platform-Based MPEG-4 Advanced Video Coding (AVC) Decoder with Block Level Pipelining," ("*Wang*"), by Wang et al. in *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*, Vol. 1, pp. 51-55, that I obtained from the

Google Exhibit No. 1063
Page 0027

Linda Hall Library. When I began researching the *Wang* article, I searched WorldCat for libraries that held *ICICS-PCM 2003* that contains *Wang*, and a record at https://search.worldcat.org/title/71496403 showed that Linda Hall Library was one of the holding libraries. After verifying their holdings information in their online catalog, I requested a copy of *Wang* from the library. This library copy is presented as **Ex. 1007** in this declaration.

44. Page 1 is internal page 51 that shows the title, authors, and abstract of *Wang*. The top of the page shows this article is part of the "ICICS-PCM2003" conference that took place from "15-18 December 2003" in Singapore, and this paper was included in "1A2.6." The bottom of the page shows *Wang* has a "2003" copyright date and IEEE holds the copyright. It also shows *Wang* is included in a book whose ISBN (International Standard Book Number) is "0-7803-8185-8." **Ex. 1007** shows *Wang* has a total of five page, including eight figures, four tables, and ten references.

45. Page 6 is the cover that shows "ICICS-PCM 2003, 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and Fourth Pacific-Rim Conference on Multimedia, 15-18 Dec 2003" and indicates this copy is Volume 1 of the conference proceedings. The cover also shows "Technical sponsorships by: IEEE Singapore Communications Chapter [and] IEEE Singapore Signal Processing Chapter." Page 7 is the title page that shows the same conference information as the cover and includes the location of the

Google Exhibit No. 1063
Page 0028

conference. Page 8 is the copyright page that shows a hand-written call number of "TK5102.9 .I62 2003 v. 1 quarto" and identifies this publication as "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)" that shows a "2003" copyright date, with the Institute of Electrical and Electronics Engineers, Inc. as the copyright holder. This page also shows "03EX758" as the IEEE catalog number, "0-7803-8185-8" as the ISBN, and "2003111734" as the Library of Congress control number. Page 9 shows the conference organizer, two technical sponsors, and Microsoft Research Asia and Singapore Exhibition and Convention Bureau as two conference sponsors. The left side of this page shows a date notation of "6/16/2004" that indicates this conference proceedings was added to the Linda Hall Library collections on June 16, 2004. The knowledge of this date annotation practice is based on my personal experience with the Linda Hall Library and their collections. The table of contents of Volume 1 begins on page 13 (internal page vii) and ends on page 27 (internal page xxi), and page 14 (internal page viii) shows *Wang* begins on page 51 and is identified as "1A2.6 P0330" that indicates the article is included in session "1A2 – Video Compression I."

46. To authenticate **Ex. 1007**, I searched for *Wang* in Google Scholar and a record led me to the IEEE metadata record for this article at https://ieeexplore.ieee.org/abstract/document/1292411. I then used the PDF link to

Google Exhibit No. 1063
Page 0029

obtain a copy of *Wang* from IEEE. This publisher copy is presented as **Ex. 1007A** in this declaration.

47. **Ex. 1007A** is a true and correct copy of *Wang*. Page 1 (internal page 51) shows the title, authors, and abstracts. This exhibit shows *Wang* has a total of five pages, including eight figures, four tables, and ten references. I have closely compared **Ex. 1007** and **Ex. 1007A** and found them to contain the same content for *Wang*.

48. **Ex. 1007** is in a condition that creates no suspicion about its authenticity. Specifically, this copy is not missing any intermediate pages, and there are no visible alterations to the document. Furthermore, the copy of *Wang* (**Ex. 1007**) has the same content as **Ex. 1007A**, which is in the custody of a well-known public research library. Based on my review and analysis, it is my opinion that **Ex. 1007** is authentic.

B. IEEE Metadata Record

49. To determine when *Wang* became publicly accessible on the Internet, I searched IEEE Xplore for records. **Ex. 1007B** is a true and correct copy of the metadata record for *Wang* that I located at https://ieeexplore.ieee.org/abstract/document/1292411 on IEEE Xplore. I personally located, identified, and obtained this record. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

50. Page 1 shows the title, authors, abstract and publisher of *Wang*. It also shows *Wang* was cited 14 times in papers and 23 times in patents. The "Published in" field shows *Wang* appears in the "Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint" and the DOI (digital object identifier) of *Wang* is "10.1109/ICICS.2003.1292411." The "Date Added to IEEE Xplore" shows the metadata record for *Wang* was added to IEEE Xplore on "04 May 2004." This page also shows the ISBN (International Standard Book Number) of the conference proceedings containing *Wang* is "0-7803-8185-8."

51. **Ex. 1007B** shows that the metadata record for *Wang* was added to IEEE Xplore on May 4, 2004. Since IEEE Xplore is available to the general public for searching, and subscribers to IEEE Xplore can freely access full text articles in this system, it is my opinion that *Wang* became publicly discoverable and accessible on IEEE Xplore from that date on.

52. As evidence that IEEE Xplore, the digital library of IEEE, was accessible to the general public and subscribers when the metadata record for *Wang* became available in 2004, I obtained the IEEE Xplore website archived by Internet Archive in 2001. I personally located, identified, and obtained this archive copy. It is presented as **Ex. 1007C** in this declaration.

53. Internet Archive is a digital library that archives scholarly materials,

Google Exhibit No. 1063
Page 0031

cultural artifacts, and other web resources on the Internet and provides the public with free access to its collections. I have used the Internet Archive in the past and am familiar with their practice of assigning an extended URL to an archived resource in the format of http://www.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] to show the point of time the archived file was captured. Since Internet Archive usually does not archive web resources the moment they appear on the Internet, and a web resource must be available first before it can be archived, the archive date of a file captured by the Internet Archive is typically later than the date when the file first became publicly accessible. In other words, the archive date means the archived file was typically available before, and definitely no later than, the archive date.

54. Page 1 of **Ex. 1007C** is a screen capture of a Wayback Machine calendar that shows the IEEE Xplore website (http://ieeexplore.ieee.org/lpdocs/epic03/) was archived 115 times between "7 Apr 2000" and "2 Dec 2024" and this particular copy was archived on "FEB 07 2001" and available at https://web.archive.org/web/20010207003457/http://ieeexplore.ieee.org/lpdocs/epic03/. Page 2 is the PDF of this archived copy that explains that "IEEE Xplore provides full-text access to IEEE transactions, journals, magazines and conference proceedings published since 1988 and all current IEEE standards." Under "Free to

Google Exhibit No. 1063
Page 0032

All," the archived copy shows the general public can have "[a]ccess to tables of contents of IEEE transactions, journals, magazines, conference proceedings and standards." The archived copy also explains that users affiliated with corporate, government and university subscribers "can search and access abstract/citation records, and full-text documents of the IEEE online publications to which" the institutions subscribe.

55. **Ex. 1007C** informs my opinion that IEEE Xplore was available to the general public for searching since at least 2001, and individual subscribers and users affiliated with institutional subscribers have been able to access full text of documents freely. Non-subscribers have been able to purchase the full text of documents after they have identified abstract/citation records in IEEE Xplore.

C. Copyright Registration Record

56. **Ex. 1007D** is a true and correct copy of the copyright registration record for *ICICS-PCM 2003* that I personally located, identified, and obtained from the public catalog of the US Copyright Office. The title, description, copyright claimant, and other titles match the information presented in **Ex. 1007**, confirming this record represents *ICICS-PCM 2003*. The "Date of Publication" field shows this conference proceedings was published on "2003-12-15" (*i.e.*, December 15, 2003), and the "Registration Number / Date" field shows this work was registered with the US Copyright Office on "2004-10-15" (*i.e.*, October 15, 2004), with a registration

Google Exhibit No. 1063
Page 0033

number of "TX0006050947."

D. Records of Linda Hall Library

57. To determine the public availability date of *Wang* at libraries, I obtained records for *ICICS-PCM 2003* that contains *Wang* from the online catalog of the Linda Hall Library. **Ex. 1007E** is a true and correct copy of the bibliographic and MARC records for *ICICS-PCM 2003*. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

58. Pages 1 to 4 of **Ex. 1007E** are the bibliographic record that shows the title of this work and indicates this title is available at "Linda Hall Library Books – 2nd Floor (TK5102.9 .I62 2003)." The record details on page 3 shows the title, subtitles, sponsors, other title, contributing organizations, subjects, ISBNs, OCLC number, publisher, creation date ("c2003"), format ("3 v."), IEEE catalog number for this work, and a note of bibliographical references and indexes. The title, subtitles, sponsors, and ISBN match the information presented in **Ex. 1007**, confirming this record represents *ICICS-PCM 2003*.

59. Page 5 is a copy of the MARC record. The first six digits of Field 008 show the MARC record was created on "040616" (*i.e.*, June 16, 2004), and the "s2003" code following the record creation date indicates this work has a single publication date of 2003. Field 005 is a machine-generated field that records the last

Google Exhibit No. 1063
Page 0034

time the Linda Hall Library modified their MARC record, and it shows the last record modification date was "20040616145300" (*i.e.*, June 16, 2004, at 2 p.m. and 53 minutes). Subfields "a" and "c" of Field 040 shows "LHL" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "LHL" is the OCLC library symbol for the Linda Hall Library. The "6/16/2004" date annotation in **Ex. 1007** shows *ICICS-PCM 2003* was added to the Linda Hall Library collections on June 16, 2004. This date stamp and the data from Fields 008, 005, and 040 inform my opinion that the Linda Hall Library created an original MARC record for the conference proceedings containing *Wang* on June 16, 2004, and added this publication to their collections on that day, making the conference proceedings discoverable in their online catalog from that date on. A customary library practice is to process a newly cataloged work promptly for public access, usually within a week after the cataloging work is completed. It is therefore my opinion that *ICICS-PCM 2003* that contains *Wang* became publicly discoverable at the Linda Hall Library on June 16, 2004. After processing, the physical copy would become publicly accessible by June 23, 2004.

60. Field 020 shows the ISBN, Field 111 shows the conference name is used as the corporate author of this work, Field 245 shows the title, subtitles and sponsors, and Field 246 shows a variant title is included to help users discover this

Google Exhibit No. 1063
Page 0035

conference proceedings. Field 260 shows the publisher information, Field 300 shows this work has three volumes. Field 500 shows the IEEE catalog number for this proceeding ("03EX758"), and Field 504 shows this proceeding includes bibliographical references and indexes. Fields 710 show the conference organizer and two technical sponsors are included to help users discover this work. Fields 711 shows the two conferences that contributed to *ICICS-PCM 2003* are also included to help the discovery of this work.

61. The subjects of this conference proceedings are represented in Field 090 by a Library of Congress Classification (LCC) number of "TK5102.9" that represents periodicals on "Signal processing" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Fields 650 by four Library of Congress subject headings, with "Signal processing," "Wireless communication systems," "Multimedia systems," and "Information technology" as the main headings, each followed by "Congresses" as a form subheading encoded in subfield "v" ("|v") to indicate the main topics are treated at a conference. These subject representations enable users to conduct subject searches to discover the conference proceedings that contains *Wang*.

62. This MARC record (**Ex. 1007E**) shows that the Linda Hall Library created an original record for the conference proceedings that contains *Wang* on June

Google Exhibit No. 1063
Page 0036

16, 2004, and last modified the record on the same day. These data inform my opinion that the conference proceedings that contains *Wang* became discoverable at the Linda Hall Library on June 16, 2004. As of that date, interested users could search for the conference proceedings that contains *Wang* by ISBN, conference name, title, variant title, conference organizer, technical sponsors, and two conference names. They could also conduct subject searches to discover this work.

63. The date annotation in **Ex. 1007** shows the conference proceedings that contains *Wang* was added to the collections of the Linda Hall Library on June 16, 2004. This data and the MARC record dates inform my opinion that the conference proceedings that contains *Wang* was publicly discoverable from June 16, 2004 on, and the physical copy, after processing, would become publicly accessible by June 23, 2004.

64. As evidence that the Linda Hall Library had an online catalog in operation in 2004 when the library added the conference proceedings containing *Wang* to their collections, I obtained a profile of the library at https://librarytechnology.org/library/2294 from libraries.org, a directory of libraries throughout the world. I personally located, identified, and obtained this profile. This is the type of record experts in my field would reasonably rely upon when forming their opinions. This profile is presented as **Ex. 1007F** in this declaration.

65. The Technology Profile on page 2 of **Ex. 1007F** shows the Linda Hall

Google Exhibit No. 1063
Page 0037

Library began using Horizon as their online catalog in 1994, and changed to Voyager in 2010. These dates mean that in 2004 interested users were able to use the Horizon system to search for the conference proceedings containing *Wang* and access this article at the Linda Hall Library.

E. Summary of My Opinion on *Wang*

66. The copy of *Wang* obtained from the Linda Hall Library (**Ex. 1007**) is authentic because it contains no signs to raise suspicion about its authenticity. In addition, this library copy has the same content as the copy obtained from IEEE, the publisher (**Ex. 1007A**).

67. The IEEE metadata record (**Ex. 1007B**) shows this index record was added to IEEE Xplore, the digital library of IEEEE, on May 4, 2004. Since IEEE Xplore is available to the general public for searching, as of May 4, 2004, interested users would be able to use IEEE Xplore to discover *Wang*. Interested users who were personal subscribers to IEEE Xplore or were related to institutional subscribers would be able to freely access the full text of documents (including *Wang*) on IEEE Xplore. Interested users without subscriptions could still use this system to discover *Wang* and purchase the article from IEEE.

68. The copyright registration record for *ICICS-PCM 2003* (**Ex. 1007D**) shows a publication date of December 15, 2003. The MARC record for this conference proceedings that contains *Wang* (**Ex. 1007E**) shows the Linda Hall

Google Exhibit No. 1063
Page 0038

Library created an original record for this work on June 16, 2004, and added it to their collections on the same day, making this work discoverable in their online catalog from that date on. The physical copy, after processing, would become publicly accessible by June 23, 2004.

69. Based on all the records and documents I have reviewed, it is my opinion that the conference proceedings of *ICICS-PCM 2003* that contains *Wang* became discoverable and accessible on IEEE Xplore on May 4, 2004. This work became discoverable at the Linda Hall Library on June 16, 2004, and the physical copy, after processing, would become publicly accessible by June 23, 2004.

V. Authenticity and Public Availability of *Wolters* (Ex. 1041)

A. Authentication

70. **Ex. 1041** is a true and correct copy of "A closer look into MPEG-4 High Efficiency AAC," ("*Wolters*"), by Wolters et al., identified as Convention Paper 5871 in *Audio Engineering Society 115th Convention, 2003 October 10-13, New York, NY*, that I obtained from the Linda Hall Library. This copy is presented as **Ex. 1041** in this declaration.

71. Pages 1 to 16 are the full text of *Wolters*. Page 1 shows the title, authors, and abstract of *Wolters*, and the top of the page shows this article is Convention Paper 5871 "presented at the 115th Convention" of the Audio Engineering Society that took place in "2003 October 10-13" in "New York, NY, USA." A note explains that "[t]his

Google Exhibit No. 1063
Page 0039

convention paper has been reproduced from the author's advance manuscript" and "[a]dditional papers may be obtained by sending request and remittance to Audio Engineering Society" in New York. A note at the bottom of internal pages 2 to 16 indicates *Wolters* is part of the "AES 115TH Convention, New York, NY, USA, 2003 October 10-13." **Ex. 1041** shows *Wolters* has a total of 16 pages in six sections, including 15 figures, four tables, and 35 references.

72. To authenticate **Ex. 1041**, I searched for *Wolters* in Google Scholar and a record led me to the record for *Wolters* (https://aes2.org/publications/elibrary-page/?id=12456) at the AES E-Library. I then followed the instructions to obtain the full text of *Wolters* from AES. This AES copy of *Wolters* is presented as **Ex. 1041A** in this declaration.

73. **Ex. 1041A** is a true and correct copy of *Wolters*. Page 1 shows the title, authors, and abstract of *Wolters*, and the top of the page shows this article is Convention Paper 5871 "presented at the 115th Convention" of the Audio Engineering Society that took place in "2003 October 10-13" in "New York, NY, USA." A note at the bottom of internal pages 2 to 16 indicates *Wolters* is part of the "AES 115TH Convention, New York, NY, USA, 2003 October 10-13." This exhibit shows *Wolters* has a total of 16 pages in six sections, including 15 figures, four tables, and 35 references. I have closely compared **Ex. 1041** and **Ex. 1041A** and found them to contain the same content for *Wolters*.

74. **Ex. 1041** is in a condition that creates no suspicion about its authenticity. Specifically, this copy is not missing any intermediate pages, and there are no visible alterations to the document. Furthermore, the copy of *Wolters* (**Ex. 1041**) has the same content as **Ex. 1041A**, which is in the custody of the Audio Engineering Society that is the organizer of AES Conventions and provides convention papers for sale to interested users. Based on my review and analysis, it is my opinion that **Ex. 1041** is authentic.

B. Records of Linda Hall Library

75. To determine the public availability date of *Wolters* at libraries, I obtained records for "Convention papers: presented at the 115th Convention: 2003, October 10-13, New York, N.Y." that contains *Wolters* from the online catalog of the Linda Hall Library. **Ex. 1041B** is a true and correct copy of the bibliographic and MARC records for this work. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

76. Pages 1 to 3 of **Ex. 1041B** are the bibliographic record that shows the title of this publication in brackets, indicating this is a cataloger-supplied title because this publication does not have a title page. The record shows this publication is available at "Linda Hall Library Closed Stacks - Books (TK7881.4 .A9235 115th 2003)." The "Location Items" field on page 2 shows this publication contains three

Google Exhibit No. 1063
Page 0041

volumes. The record details on page 2 shows the cataloger-supplied title, subtitles, creator/contributors, subjects, OCLC record number for this record ("56052226"), publisher information, creation date ("2003"), format (a total of [139] papers), and general notes that indicate the title is based on the caption and that the publication includes papers nos. 5854-5993 with omissions.

77. Page 4 is a copy of the MARC record. The first six digits of Field 008 show the MARC record was created on "040729" (*i.e.*, July 29, 2004), and the "s2003" code following the record creation date indicates this work has a single publication date of 2003. Subfields "a" and "c" of Field 040 shows "LHL" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "LHL" is the OCLC library symbol for the Linda Hall Library. In the process of obtaining the Linda Hall Library copy of Wolters (**Ex. 1041**), I learned from the library that *Convention Papers: Presented at the 115th Convention* does not come with a title page or a table of contents, and each preprint is only available without introductory information. Data from Fields 008 and 040 inform my opinion that the AES 115th Convention Papers that contains *Wolters* became discoverable at the Linda Hall Library on July 29, 2004, when the library created their original MARC record. A customary library practice is to process a newly cataloged work promptly to make it publicly accessible, usually within a week after the cataloging work is

completed. It is therefore my opinion that, after processing, the physical copy of the AES 115th Convention Papers (and *Wolters* included therein) would become publicly accessible at the Linda Hall Library by August 5, 2004.

78. Field 110 shows the conference name is used as the corporate author of this work, Field 245 shows the information provided by the cataloger for the title, subtitles and corporate body, and Field 260 shows the publisher information in brackets to indicate the information is provided by the cataloger. Field 300 shows this work has a total of 139 pages, and the bracketed page number indicates the cataloger provided this page count number at the time of cataloging. Fields 500 explains the cataloger-supplied title is based on the caption, and the publication includes paper nos. 5854-5993. Field 710 shows the Audio Engineering Society corporate name was included to help users discover this work.

79. The subjects of this work are represented in Field 090 by a Library of Congress Classification (LCC) number of "TK7881.4" that represents general works on "Sound systems. Sound recording. Sound reproduction" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Fields 650 by two Library of Congress subject headings, with "Electro-acoustics" and "Acoustical engineering" as the main headings, each followed by "Congresses" as a form subheading encoded in subfield "v" ("|v") to indicate the main topics are

treated at a conference. These subject representations enable users to conduct subject searches to discover the publication that contains *Wolters*.

80. This MARC record (**Ex. 1041B**) shows that the Linda Hall Library created an original record for the AES 115th Convention Papers that contains *Wolters* on July 29, 2004, and last modified the record on February 14, 2005. These data, together with the date stamps included in **Ex. 1041**, inform my opinion that the AES 115th Convention Papers that contains *Wolters* became discoverable at the Linda Hall Library on July 29, 2004. As of that date, interested users could search for this work by the conference name, title, and the Audio Engineering Society. They could also conduct subject searches to discover this work.

81. As evidence that the Linda Hall Library had an online catalog in operation in 2004 when the original record for the AES 115th Convention Papers that contains *Wolters* was created, I obtained a profile of the library at https://librarytechnology.org/library/2294 from libraries.org, a directory of libraries throughout the world. I personally located, identified, and obtained this profile. This is the type of record experts in my field would reasonably rely upon when forming their opinions. This profile is presented as **Ex. 1041C** in this declaration.

82. The Technology Profile on page 2 of **Ex. 1041C** shows the Linda Hall Library began using Horizon as their online catalog in 1994, and it changed to Voyager in 2010. These dates mean that in 2004 interested users were able to use the

Google Exhibit No. 1063
Page 0044

Horizon system to search for the publication containing *Wolters* and access this article at the Linda Hall Library.

C. Records of the Canadian National Science Library

83. My research on the public availability of the AES 115th Convention Papers that contains *Wolters* at libraries also discovered that the Canadian National Science Library made this work publicly accessible in late 2003. **Ex. 1041D** is a true and correct copy of the bibliographic and MARC records for *Convention Papers: Presented at the 115th Convention*. I personally located, identified, and obtained these records from the Canadian National Science Library. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

84. Pages 1 and 2 are the bibliographic record that shows the title, publisher, call numbers ("TK5981 A91 2003 Oct no. 5854-5920" and "TL5981 A91 2003 Oct No. 5921-5993"), location ("NRC Ottawa," which is the National Science Library at Ottawa), and status of this work ("In library use only). The corporate author, title, publisher match the information presented in **Ex. 1041**, confirming this record represents the AES 115th Convention Papers that contains *Wolters*.

85. Pages 3 and 4 are the MARC record. The first six digits of Field 008 show the record was created on "031212" (*i.e.*, December 12, 2003), and the "s2003" code after the record creation date indicates this work has a single publication date of 2003. Subfields "a" and "c" of Field 040 show "Ca00N" created the original record

and entered it into the OCLC WorldCat database. According to the record for the National Science Library of Canada (https://sigles-symbols.bac-lac.gc.ca/eng/Search/Details?Id=1515) in the Canadian Library Directory, "00N" is the Canadian Library symbol for the National Science Library. Data in Field 008 and Field 040 inform my opinion that the National Science Library of Canada created an original record for the AES 115th Convention Papers that contains *Wolters* on December 12, 2003, making this work publicly discoverable in their online catalog from that date on. Since then, interested users have been able to discover this work and access it at the library.

86. Field 110 shows the conference name is used as the corporate author of this work. Field 245 shows the title in brackets and Field 260 shows the publisher and publication year in brackets, and Field 500 explains that the cataloger provided the information at the time of record creation. Field 710 shows the Audio Engineering Society is included in the MARC record to help users discover this work.

87. The subjects are represented in Field 090 by a Library of Congress Classification number of "TK5981" that represents general works on "Electroacoustics. Electroacoustic transducers" (source: Library of Congress Classification, Schedule T at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Fields 650 by two Library of Congress subject headings, with "Electro-acoustics"

and “Acoustical engineering” as the main topics, each followed by “Congresses” as a form subheading encoded in subfield “v” (“|v”) to indicate the main topics are treated at a conference. These subject representations support users to conduct subject searches to discover this work.

88. This MARC record (**Ex. 1041D**) has made “Convention Papers: Presented at the 115th Convention, 2003 October 10-13, New York, New York” of the Audio Engineering Society discoverable since the library created the MARC record for it on December 12, 2003. As of that date, interested users could discover this work by the conference name, the title, and the Audio Engineering Society. In addition, interested users could conduct subject searches by the Library of Congress Classification number, and the Library of Congress subject headings in this record to discover this work.

89. As discussed earlier, original records entered into the OCLC WorldCat database become available to OCLC members for cataloging, interlibrary loan services, and collection management services. In addition, the records become discoverable in WorldCat, the free web portal at worldcat.org, which is freely available to the general public on the Internet. The records also become discoverable in OCLC FirstSearch, which is a search platform for subscribers to access many proprietary databases. The difference between WorldCat, the free web portal, and OCLC FirstSearch is that the latter requires subscription for access and its records

Google Exhibit No. 1063
Page 0047

contain more technical details, such as the date of record entry in the "Entry" field.

90. As evidence that the original record for the AES 115th Convention Papers that contains *Wolters* created by the National Science Library of Canada became available on OCLC FirstSearch in 2003, I obtained the FirstSearch record and attached it to **Ex. 1041D** as pages 5 and 6.

91. The top of the page 5 shows the title, author, and accession number of this FirstSearch record, and also shows the National Science Library of Canada, whose OCLC library symbol is "CAI," holds this title. The "Entry" field, the database field, and the Accession number field on page 6 show the record was entered into the OCLC WorldCat database on "20031212" (*i.e.*, December 12, 2003), and its OCLC number is 927388398." This FirstSearch record confirms the National Science Library of Canada created their original record for the AES 115th Convention Papers that contains *Wolters* and entered it into the OCLC WorldCat database on December 12, 2003, making this record available from that date on to OCLC members, the general public on WorldCat, the free web portal, and subscribers to OCLC FirstSearch.

92. To summarize, library records from the National Science Library of Canada show the AES 115th Convention Papers that contains *Wolters* became discoverable on December 12, 2003, to their library users, to OCLC members, to FirstSearch subscribers, and to the general public who have access to WorldCat, the

Google Exhibit No. 1063
Page 0048

free web portal. In addition, the library records of the Linda Hall Library show this work became discoverable at this library by July 29, 2004, and the physical copy, after processing, would become publicly accessible by August 5, 2004.

D. Internet Archive Records

93. Internet Archive is a digital library that archives scholarly materials, cultural artifacts, and other web resources on the Internet and provides the public with free access to its collections. I have used the Internet Archive in the past and am familiar with their practice of assigning an extended URL to an archived resource in the format of http://www.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] to show the point of time the archived file was captured. Since Internet Archive usually does not archive web resources the moment they appear on the Internet, and a web resource must be available first before it can be archived, the archive date of a file captured by the Internet Archive is typically later than the date when the file first became publicly accessible. In other words, the archive date means the archived file was typically available before, and definitely no later than, the archive date.

94. To determine the public availability date of *Wolters* on the Internet, I searched Internet Archive for records. I personally located, identified, and obtained these records. They are the type of record experts in my field would reasonably rely upon when forming their opinions.

95. **Ex. 1041E** is a true and correct copy of the 115th AES Convention schedule for "Session C Low Bit-Rate Audio Coding" on Friday, October 10. Page 1 is a screen capture of a Wayback Machine calendar that shows this schedule was archived 20 times between "31 Aug 2003" and "13 Apr 2024" and this particular copy was archived on "AUG 31 2003" and available at https://web.archive.org/web/20030831165223/http://www.aes.org/events/115/papers/SessionC.cfm. Pages 2 and 3 are the PDF of this archived copy that shows *Wolters* is "C-4 A Closer Look into MPEG-4 High Efficiency AAC."

96. Pages 4 to 6 show the schedule was also archived on December 17, 2003, showing the same information on *Wolters*. Archived copies in **Ex. 1041E** inform my opinion that interested users could discover the schedule of Session C and learn about the presentation of *Wolters* by August 31, 2003, and by December 17, 2003. If interested, they could then contact the Audio Engineering Society to obtain the preprint of *Wolters*.

97. As evidence that the preprint of *Wolters* was available for purchase, I obtained an archived copy of the webpage for "AES Special Publications." I personally located, identified, and obtained this copy. This is the type of material experts in my field would rely upon when forming their opinions.

98. **Ex. 1041F** is a true and correct copy of "AES Special Publications" archived on "FEB 18 2004" and available at

https://web.archive.org/web/20040218195627/http://aes.org/publications/convention_cds.cfm. Page 1 is a screen capture of a Wayback Machine calendar showing the archiving history of this webpage. Pages 2 and 3 are the PDF of the archived copy. Under "Convention Preprints," the webpage informs visitors that "[s]ingle Convention Preprints are available through the AES Preprint Search and Shop facility. At the bottom of this page you can find complete sets." The list on page 2 shows that by February 18, 2004, interested users could purchase the full preprint set for the 115th Convention on CD-ROM for $110 (for members) or $160 (for others). **Ex. 1041F** informs my opinion that by February 18, 2004, interested users could obtain *Wolters*, because it is included in the preprints for the 115th Convention.

99. My research on Internet Archive also discovered that individual preprints of the AES 115th Convention were available for purchase. **Ex. 1041G** is a true and correct copy of the list of preprints for the "AES 115th Convention" that is part of "Audio Engineering Society Preprints." Page 1 is a screen capture of a Wayback Machine calendar that shows the archiving history of this list, and indicates this particular copy was archived on "JUN 15 2004" and available at https://web.archive.org/web/20040615041014/http://aes.org/publications/preprints/lists/115.cfm. Pages 2 to 27 are the "Preprints Listing" for the AES 115th Convention starting with "5854" and ending with "5993" and page 2 informs interested users that "[s]ingle Convention Preprints are available through the AES Reprint Search and

Google Exhibit No. 1063
Page 0051

Shop facility." Page 5 shows the authors, abstract, and title of *Wolters* under "5871."

100. As evidence that preprints for the AES 115th Convention could be searched and purchased by early 2004, I obtained the webpage for "Audio Engineering Society Preprints" archived on "APR 04 2004" and attached this archived copy as pages 28 and 29 to **Ex. 1041G**. Page 28 is a screen capture of a Wayback Machine calendar that shows the archiving history of this webpage and indicates this particular copy was archived on April 4, 2004, and available at https://web.archive.org/web/20040404224624/http://www.aes.org/publications/preprints/search.html. Page 29 is the PDF of this archived copy that shows a search form and informs interested users that "[t]his search engine provides querying through a database of AES Preprints for AES Conventions **1** through **115**; by Author, Title, Preprint Number, Associated Convention Number, Associated Convention date, Journal Volume, Journal Issue and Year of Journal Publication (if applicable)." The webpage further explains that the search system "will return matching results providing all available information and includes an option to automatically generate a preprint order form for one or more chosen articles." It also informs interested users that they may "purchase full preprint sets for the 104th through the 115th Convention on CD-ROM and in paper format from Special Publications." This webpage further affirms that "The AES Preprints database is now complete through convention #115."

101. Taken together, **Ex. 1041E**, **Ex. 1041F** and **Ex. 1041G** inform my

opinion that by August 31 and December 17 of 2003, interested users would have been able to discover the *Wolter* presentation at the AES 115th Convention; that by February 18, 2004, interested users would discover they could purchase the full preprint set for the 115th Convention on CD-ROM, and use it to access *Wolters*; and that by April 4, 2004, interested users would know they could use the AES preprint search engine to discover *Wolters* and order a single preprint of this article from AES.

E. Usage Records

102. Actual usage of a publication is reflected by other publications that make reference to it. I have reviewed the citation history of *Wolters* on Google Scholar to compile a list of early citations. **Ex. 1041H** is a list of three citations to *Wolters* that were published in June and December of 2004, supporting my opinion that *Wolters* was publicly discoverable and accessible by February 18, 2004, from the Audio Engineering Society Preprint.

F. Summary of My Opinion on *Wolters*

103. The copy of *Wolters* (**Ex. 1041**) is authentic because it contains no signs to raise suspicion about its authenticity. In addition, this copy has the same content as the copy obtained from the Audio Engineering Society (**Ex. 1041A**).

104. The National Science Library of Canada created a record for the AES 115th Convention Papers that contains *Wolters* on December 12, 2003 (**Ex. 1041C**) and entered the record into the OCLC WorldCat database, making this work

discoverable from that date on for their library users, OCLC members, FirstSearch subscribers, and users of WorldCat, the free web portal. In addition, the Linda Hall Library created their record on July 29, 2004 (**Ex. 1041B**), making this work discoverable at this library from that date on. The physical copy, after processing, would become publicly accessible by August 5, 2004.

105. Archived copies of AES webpages show that by August 31, 2003, and December 17, 2003, interested users would have discovered that *Wolters* was included in the AES 115th Convention (**Ex. 1041E**). By February 18, 2004, interested users could purchase the full preprint set for the 115th Convention on CD-ROM, and use it to access *Wolters* (**Ex. 1041F**). By April 4, 2004, interested users could use the AES preprint search engine to discover *Wolters* and order a single preprint of this article from AES (**Ex. 1041G**). Citation history of *Wolters* on Google Scholar shows three citations appeared in June 2004 and December 2004 (**Ex. 1041H**), supporting my opinion that *Wolters* was publicly accessible in late 2003 at a library and in early 2004 on the Internet.

106. Based on the records and documents I have reviewed, it is my opinion that *Wolters* was publicly accessible from the Audio Engineering Society webpages by February 2004 as part of a full set of preprints, and by April 2004 as a single preprint. Public access to the AES 115th Convention Papers became publicly discoverable by December 12, 2003, at the National Science Library of Canada, and

Google Exhibit No. 1063
Page 0054

became publicly discoverable at the Linda Hall Library on July 29, 2004.

VI. Public Availability of *Watkinson* (Ex. 1005)

107. **Ex. 1005** is a true and correct copy of excerpts of *The MPEG Handbook: MPEG-1, MPEG-2, MPEG-4*, ("*Watkinson"*), by John Watkinson, published by Focal Press, that I received from counsel. This copy is presented as **Ex. 1005** in this declaration.

108. Page 1 is the cover that shows the author, title, subtitle, and publisher. Page 2 is the title page that shows the same information as the cover and indicates the publisher's locations. Page 3 is the copyright page that shows Focal Press is an imprint of Butterworth-Heinemann of Oxford, and this book was first published in 2001 with a 2001 copyright date and the author holds the copyright. This page shows the book's ISBN (International Standard Book Number) is "0 240 51656 7" and the British Library and the Library of Congress produced "Cataloguing in Publication Data" (*i.e.*, pre-publication records) for this work. The table of contents appears from page 4 (internal page v) to page 8 (internal page ix) and shows *Watkinson* includes seven chapters, bibliographical references and an index.

A. Library of Congress Records

109. To determine the public availability date of *Watkinson*, I obtained Library of Congress records for this work. **Ex. 1005A** is a true and correct copy of the bibliographic and MARC records for *Watkinson*. I personally identified, located,

Google Exhibit No. 1063
Page 0055

and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

110. Pages 1 and 2 of **Ex. 1005A** are the bibliographic record that shows the author, title, subtitle, publisher, and publication year of this work. It shows the ISBN ("0240516567"), subjects, note on bibliographical references and index, and other details. The author, title, publisher, publication year, ISBN and note match the information presented in **Ex. 1005**, confirming this record represents *Watkinson*. The "Item Availability" area shows the call number of *Watkinson* is "TK6680.5 .W3843 2001" and users can request access to copy 1 at the "Jefferson or Adams Building Reading Rooms" and the book's status is "Not Charged" (meaning available for user request).

111. Pages 3 and 4 are the MARC record for *Watkinson*. The first six digits of Field 008 show the MARC record was created on "010827" (*i.e.*, August 27, 2001). Field 955 records local cataloging activities and shows the library created a pre-publication record on "2001-08-27" (*i.e.*, August 27, 2001), received a copy of the published work on "2001-11-07" (*i.e.*, November 7, 2001) for "CIP ver." (*i.e.*, record verification at the Cataloging in Publication Program), and completed the cataloging work and sent the book to BCCD (the Binding and Collection Care Division) for processing on "2001-11-21" (*i.e.*, November 21, 2001). Subfields "a" and "c" of Field

Google Exhibit No. 1063
Page 0056

040 shows that "DLC" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library symbol for the Library of Congress. Fields 008, 955 and 040 inform my opinion that Library of Congress created an original MARC record for *Watkinson* on August 27, 2001 before the book was published, received the book for CIP record verification on November 7, 2001, and sent the book to the processing unit on November 21, 2001. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that *Watkinson* became publicly accessible at the Library of Congress no later than February 21, 2002.

112. Field 010 of the MARC record shows the Library of Congress Control Number and Field 020 show the ISBN of this book. Field 100 shows the author, Field 245 shows the title, subtitle and statement of responsibility, and Field 260 shows the publisher and publication year. Field 300 shows the book's front matter ends on page ix, the last numbered page is page 395, and the book includes illustrations. Field 504 shows *Watkinson* includes bibliographical references and index.

113. Subjects of this book are represented in Field 050 by a Library of

Congress Classification (LCC) number of "TK6680.5" that represents general works on "Digital video" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of "621.388" that represents "Television" (source: Dewey Decimal Classification Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf). Subjects are also represented in Fields 650 by two Library of Congress subject headings, with "Video compression" and "MPEG (Video coding standard)" as the main topics, each followed by "Handbooks, Manuals, etc." as a form subheading encoded in subfield "v" ("|v") to indicate the main topics are treated in a handbook. These subject representations enable users to conduct subject searches to discover *Watkinson*.

114. This MARC record (**Ex. 1005A**) shows that the Library of Congress created a pre-publication record for *Watkinson* on August 27, 2001, completed their cataloging record on November 21, 2001, and sent the book to processing on the same day. These dates inform my opinion that since November 21, 2001, *Watkinson* has become discoverable in the online catalog of the Library of Congress. As of that date, interested users could conduct subject searches by the LCC number, DDC number and the Library of Congress subject headings in the MARC record to discover this book. They could also search for this book by the author, title, and ISBN.

B. Summary of My Opinion on *Watkinson*

115. Based on the document and records I have reviewed and analyzed, it is my opinion that *Watkinson* has been publicly discoverable in the online catalog of the Library of Congress since November 21, 2001. The physical copy, after processing, would become publicly accessible at the Library of Congress by February 21, 2002.

VII. Public Availability of *Gourley* (Ex. 1006)

116. **Ex. 1006** is a true and correct copy of excerpts of *HTTP: the Definitive Guide*, ("*Gourley*"), by David Gourley and Brian Totty, published by O'Reilly, that I received counsel. This copy is presented as **Ex. 1006** in this declaration.

117. Page 1 is the cover that shows the authors, title, subtitle, and publisher. Page 2 is the title page that shows the same information as the cover and indicates the publisher's offices. Page 3 is the copyright page that shows O'Reilly Media, Inc. of Sebastopol, CA, published this "First Edition" in September 2002, and the book has a "2002" copyright date with the publisher as the copyright holder. This page also shows "978-1-56592-509-0" as the ISBN (International Standard Book Number) of *Gourley*. The table of contents appears from page 4 (internal page v) to page 10 (internal page xi) and shows that *Gourley* contains 21 chapters in five parts, Part VI includes eight appendices, and the index begins on page 617.

Google Exhibit No. 1063
Page 0059

A. Library of Congress Records

118. To determine the public availability date of *Gourley*, I obtained Library of Congress records for this work. **Ex. 1006A** is a true and correct copy of the bibliographic and MARC records for *Gourley*. I personally identified, located, and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

119. Pages 1 to 3 of **Ex. 1006A** are the bibliographic record that shows the author, title, subtitle, publisher, edition number, and publication year of this work. It shows the 10-digit ISBN ("1565925092"), second author, and other details. The author, title, publisher, publication year, and the 10-digit portion of the ISBN match the information presented in **Ex. 1006**, confirming this record represents *Gourley*. The "Item Availability" area shows the call number of *Gourley* is "QA76.76.H94 G695 2002," copy 1 is stored at "LANDOVR" (*i.e.*, the Landover storage facility of the Library of Congress), copy 2 is stored at "FT MEADE" (*i.e.*, the Fort Meade storage facility of the Library of Congress), users can request access to these copies at the "Jefferson or Adams Building Reading Rooms" and the status of both copies is "Not Charged" (meaning available for user request).

120. Pages 4 and 5 are the MARC record for *Gourley*. The first six digits of Field 008 show the MARC record was created on "020215" (*i.e.*, February 15, 2002).

Google Exhibit No. 1063
Page 0060

Field 955 records local cataloging activities and shows a pre-publication record was created on "2002-02-15" (*i.e.*, February 15, 2002), the book was received on "2003-01-10" (*i.e.*, January 10, 2003) for "CIP ver." (*i.e.*, record verification at the Cataloging in Publication Program), the cataloging work was completed on "2003-02-06" (*i.e.*, February 6, 2003) and the physical copy was sent to BCCD (the Binding and Collection Care Division) for processing that day. Subfields "a" and "c" of Field 040 shows that "DLC" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library symbol for the Library of Congress. Fields 008, 955 and 040 inform my opinion that Library of Congress created an original MARC record for *Gourley* on February 15, 2002, before the book was published, received the book for verification of the Cataloging in Publication record on January 10, 2003, completed the record and sent the book to the processing unit on February 6, 2003. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that *Gourley* became publicly accessible at the Library of Congress no later than May 6, 2003.

121. Field 010 of the MARC record shows the Library of Congress Control Number and Field 020 show the ISBN of this book. Field 100 shows the first author, Field 245 shows the title, subtitle and statement of responsibility, Field 250 shows the edition statement, and Field 260 shows the publisher and publication year. Field 700 shows the second author is included to help users discover this book.

122. Subjects of this book are represented in Field 050 by a Library of Congress Classification (LCC) number of "QA76.76.H94" that represents "Hypertext systems, including hypertext document markup languages" (source: Library of Congress Classification, Schedule Q, at https://www.loc.gov/aba/publications/FreeLCC/LCC_Q2020TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of "004.67/8" that represents "Wide-area networks, including Internet; World Wide Web" (source: Dewey Decimal Classification Class 000 at https://www.oclc.org/content/dam/oclc/webdewey/help/000.pdf). Subjects are also represented in Fields 650 by two Library of Congress subject headings, with "Hypertext systems," and "HTTP (Computer network protocol)" as the main topics. These subject representations enable users to conduct subject searches to discover *Gourley*.

123. This MARC record (**Ex. 1006A**) shows that the Library of Congress created a pre-publication record for *Gourley* on February 15, 2002, and completed

their cataloging record and sent the book to processing on February 6, 2003. These dates inform my opinion that *Gourley* has been discoverable in the online catalog of the Library of Congress since February 6, 2003. As of that date, interested users could conduct subject searches by the LCC number, DDC number and the Library of Congress subject headings in the MARC record to discover *Gourley*. They could also search for this book by the authors, title, and ISBN.

B. Copyright Registration Record

124. **Ex. 1006B** is a true and correct copy of the copyright registration record for *Gourley*. I personally located, identified, and obtained this record from the public catalog of the U.S. Copyright Office. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

125. The title, authors, contributors, imprint, copyright claimant, and date of creation ("2002") match the information presented in **Ex. 1006**, confirming this record represents *Gourley*. The Date of Publication field shows this book was published on "2002-09-19" (*i.e.*, September 19, 2002). These dates align well with the dates in the Library of Congress MARC record that show the library created a pre-publication record on February 15, 2002, and received a copy of the book for verification of the Cataloging-in-Publication record on January 10, 2003.

C. Summary of My Opinion on *Gourley*

126. Based on the document and records I have reviewed and analyzed, it is

my opinion that *Gourley* was published on September 19, 2002, and has been publicly discoverable in the online catalog of the Library of Congress since February 6, 2003. The physical copy, after processing, would become publicly accessible at the Library of Congress by May 6, 2003.

VIII. Public Availability of *MPEG Guide* (Ex. 1014)

127. **Ex. 1014** is a true and correct copy of A Guide to MPEG Fundamentals and Protocol Analysis (Including DVB and ATSC), ("*MPEG Guide*"), that I received from counsel. This copy is presented as **Ex. 1014** in this declaration.

128. Page 1 is the cover that shows the title and Tektronix as the source of this document. Page 2 is the title page that shows the title. Page 3 shows the title and a URL (www.tektronix.com/video_audio) of this document. Pages 5 and 6 are the table of contents that shows *MPEG Guide* has nine sections and a glossary. **Ex. 1014** shows this document has a total of 64 pages, includes figures and table, and figures are numbered within each section. Page 64 shows a copyright date of "2000" and Tektronix, Inc. holds the copyright. The "For Further Information" area on page 64 informs interested users that "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."

Google Exhibit No. 1063
Page 0064

A. Internet Archive Records

129. Internet Archive is a digital library that archives scholarly materials, cultural artifacts, and other web resources on the Internet and provides the public with free access to its collections. I have used the Internet Archive in the past and am familiar with their practice of assigning an extended URL to an archived resource in the format of http://www.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] to show the point of time the archived file was captured. Since Internet Archive usually does not archive web resources the moment they appear on the Internet, and a web resource must be available first before it can be archived, the archive date of a file captured by the Internet Archive is typically later than the date when the file first became publicly accessible. In other words, the archive date means the archived file was available before, and definitely no later than, the archive date.

130. To determine the public availability of *MPEG Guide*, I searched Internet Archive for archived information on www.tektronix.com and found that *MPEG Guide* was publicly accessible by August 7, 2001. My discovery is discussed below.

131. **Ex. 1014A** is a true and correct copy of the webpage for "A Guide to MPEG Fundamentals and Protocol Analysis, Including DVB and ATSC." I personally located, identified, and obtained this webpage. This is the type of material

experts in my field would reasonably rely upon when forming their opinions. The webpage presents information in frames, so I used screen captures to produce this exhibit.

132. Page 1 is a screen capture of a Wayback Machine calendar that shows this webpage was archived 19 times from "7 August 2001" to "15 Jul 2014" and this particular copy was archived on "AUG 07 2001" and available at https://web.archive.org/web/20010807020249/http://www.tektronix.com/Measurement/cgi-bin/framed.pl?Document=/Measurement/App_Notes/mpegfund/&FrameSet=television. Page 1 shows the title and table contents and includes a link for "Print Application Notes (pdf)." The table of contents (pages 1 to 6) shows this document has nine sections and a glossary. The headings and subheadings in **Ex. 1014A** are the same as their corresponding contents in **Ex. 1014**. These headings and subheadings plus the glossary are presented as hyperlinks in **Ex. 1014A**. Internet Archive, however, did not archive the contents under these hyperlinks. It is commonly known that Internet Archive does not archive everything on the Internet, but it did archive this webpage for *MPEG Guide* between 2001 and 2014, and the table of contents in the copy archived on August 7, 2001 (**Ex. 1014A**) is the same as that in **Ex. 1014**.

133. Page 7 shows the webpage for *MPEG Guide* archived in late 2002. The top of the page is a screen capture of a Wayback Machine calendar that shows this

copy was archived on "DEC 25 2002" and available at https://web.archive.org/web/20021225003145/http:/www.tektronix.com/Measurement/cgi-bin/framed.pl?Document=/Measurement/App_Notes/mpegfund/&FrameSet=television. The top of page 7 shows this is the webpage for *MPEG Guide* and shows "Step 1 of 2." It informs visitors to this webpage that "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." It further instructs visitors to complete a form online to download this document. Pages 8 through 10 show the form for interested users to fill out, and a "Continue" button and a "Reset" button for users to proceed to the next step. Internet Archive, however, did not archive the contents under these two links, so we are not able to see what the next step looks like. Nevertheless, pages 8 to 10 show that by December 25, 2002 interested users could follow the two steps on this webpage to download a copy of *MPEG Guide* from the Internet.

B. Summary of My Opinion on *MPEG Guide*

134. Based on the documents I have reviewed, it is my opinion that A Guide to MPEG Fundamentals and Protocol Analysis, including DVB and ATSC from tektronix was publicly accessible by August 7, 2001, on the Internet from the company's web site. By December 25, 2002, interested users could download a copy

Google Exhibit No. 1063
Page 0067

of this document after they completed two steps on the Tektronix web site.

IX. Public Availability of *Mietens* (Ex. 1018)

135. **Ex. 1018** is a true and correct copy of "New complexity scalable MPEG encoding techniques for mobile applications," ("*Mietens*"), by Mietens et al., *EURASIP Journal on Applied Signal Processing*, 2004(2), 236-252, that I received from counsel. This copy is presented as **Ex. 1018** in this declaration.

136. Page 1 (internal page 236) shows the title, authors, abstract, and keywords and phrases. The top of the page shows the source information of *Mietens* and a "2004" copyright for this article, with Hindawi Publishing Corporation as the copyright holder. This page shows the manuscript was received on "10 December 2002" and revised on "7 July 2003." **Ex. 1018** shows *Mietens* has a total of 17 pages, including 22 figures, two tables, six equations, and 18 references.

A. Record from American University

137. To determine the public availability of *Mietens*, I searched Google Scholar for records and learned that the journal that published *Mietens* was a public access journal. For public access journal articles, the publication date on the issue is the date the papers in that issue become publicly accessible on the Internet. To illustrate this discovery and access process, I have obtained the library serial record from the American University and webpages from Springer, the journal publisher. I personally located, identified, and obtained these records. They are the type of

Google Exhibit No. 1063
Page 0068

records experts in my field would reasonably rely upon when forming their opinions.

138. **Ex. 1018A** is a true and correct copy of the serial record for *EURASIP Journal on Advances in Signal Processing*, available at https://wrlc-amu.primo.exlibrisgroup.com/permalink/01WRLC_AMU/1jvcmg9/alma99186427786404102 from the online catalog of the American University Library. Page 1 shows this publication is a "peer reviewed" and "open access" journal and issues since 2001 are available online. The "View Online" area shows interested users could access articles in this journal on three platforms. Page 2 shows interested users could discover this journal by the issue body, publisher, title, variant titles, language, and notes. The "Restrictions on Access" field shows "Unrestricted online access." In addition, interested users could conduct subject searches to discover this journal, because subject terms ("Signal processing—Periodicals," "Signal processing," and "Periodicals") are include in the record.

139. The "Springer Nature OA Free Journals (WRLC)" in the library record led me to the publisher's platform at https://link.springer.com/journal/13634/volumes-and-issues. **Ex. 1018B** is a true and correct copy of webpages provided by the publisher to help interested users navigate the journal. Pages 1 to 7 show the webpage for the journal, *EURASIP Journal on Advances in Signal Processing*, and indicates on page 1 that the publishing model for this journal is "Open access." Pages 2 to 7 list volumes and issues, and page 6 shows

a link to volume 2004, issue 2 at https://link.springer.com/journal/13634/volumes-and-issues/2004-2. This link leads to a list of articles published in volume 2004, issue 2 at https://link.springer.com/journal/13634/volumes-and-issues/2004-2 (pages 8 to 11), and the link to *Mietens* leads to the webpage for *Mietens* at https://link.springer.com/article/10.1155/S1110865704309091 (page 12). The webpage for *Mietens* on page 12 shows the journal, article title, publication date "04 March 2004" and identifies *Mietens* as "Volume 2004, article number 785130" and page 15 shows the digital object identifier of *Mietens* is "https://doi.org/10.1155/S1110865704309091."

B. Summary of My Opinion on *Mietens*

140. Based on the library record and publisher webpages I have reviewed, it is my opinion that *Mietens* is an open access article that became publicly accessible on March 4, 2004 when it was published.

X. Public Availability of *Hui* (Ex. 1019)

141. **Ex. 1019** is a true and correct copy of "Client-server synchronization and buffering for variable rate multimedia retrievals," ("*Hui*"), by Hui et al., *IEEE Journal on Selected Areas in Communications*, 14(1), 226-237, that I received from counsel. This journal issue has an issue date of "January 1996." This copy is presented as **Ex. 1019** in this declaration.

142. Page 1 (internal page 226) shows the title, authors, and abstract. The

bottom shows *Hui* appears in a journal whose International Standard Serial Number (ISSN) is "0733-8716," and this article has a "1996" copyright date and IEEE (Institute of Electrical and Electronics Engineers, Inc.) holds the copyright. **Ex. 1019** shows *Hui* has a total of 12 pages in six sections that includes 13 figures, 22 equations, and 22 references.

A. Date Stamped Copy from Library of Congress

143. To determine the public availability of *Hui*, I searched WorldCat.org, the free web portal, to identify libraries holding the journal containing *Hui*, and learned that the Library of Congress was one of the holding libraries. After verifying the holdings information in their online catalog, I requested a copy of *Hui* from the Library of Congress through a document vendor. This library copy of *Hui* is presented as **Ex. 1019A** in this declaration.

144. Page 1 of **Ex. 1019A** shows the spine of a bound volume identified as "14, JAN-JUNE 1996" of the "IEEE Journal on Selected Areas in Communications" that has a call number of "TK5101.A1 I35." Page 2 is the issue cover that shows this journal is a publication of the IEEE Communications Society, its ISSN is 0733-8716 and its CODEN (a unique six- character code that identifies journals) is "ISACEM." The cover also identifies this issue as volume 14, number 1 that has a "January 1996" issue date. The table of contents appears on the cover and the back cover, and shows *Hui* begins on page 226. Page 4 is the copyright page that shows this journal is

Google Exhibit No. 1063
Page 0071

published nine times a year by the Institute of Electrical and Electronics Engineers, Inc., and this issue has a "1996" copyright date with the publisher as the copyright holder. Page 5 is the guest editorial page that carries a date stamp of "Copyright Office JAN 30 1996 Library of Congress" that indicates this issue was received by the Library of Congress on that date. Pages 6 to 17 (internal pages 226 to 237) are the full text of *Hui*.

145. I have compared **Ex. 1019** with the Library of Congress copy of *Hui* (**Ex. 1019A**) and found them to have the same content. The date stamp on **Ex. 1019A** indicates Library of Congress received this issue containing *Hui* on January 30, 1996. As discussed above, a customary library practice is to document the receipt of a new journal issue by placing the issue through the serial check-in system. After that, the new issue is promptly processed for public access, usually within a week after serial check-in is completed. It is therefore my opinion that the issue containing *Hui* became publicly accessible at the Library of Congress by February 6, 1996.

B. Library of Congress Records

146. As evidence that interested users could discover the journal holding *Hui* at the Library of Congress to access this article, I obtained serial records for *IEEE Journal on Selected Areas in Communications* from the online catalog of the Library of Congress. **Ex. 1019B** is a true and correct copy of the bibliographic and MARC records for this journal. I personally located, identified, and obtained these records.

Google Exhibit No. 1063
Page 0072

They are the type of records experts in my field would reasonably rely upon when forming their opinions.

147. Pages 1 to 4 of **Ex. 1019B** are the bibliographic record and Pages 5 to 8 are the MARC record for this journal. Field 008 of the MARC record shows the serial record was created on "820802" (*i.e.*, August 2, 1982), and the code of "c19839999" following the record creation date indicates this publication is a continuing publication that began in 1983 and has no end date for publication yet. Subfields "a" and "c" of Field 040 show "NSD (the National Serial Data Program at the Library of Congress) created the original record, and subfield "d" shows "DLC" (the Library of Congress) modified the record later. Field 022 shows the ISSN, Field 030 shows the CODEN, Field 245 shows the journal title, and Field 260 shows IEEE began publishing this journal in 1983. These details match the information presented in **Ex. 1019A**, confirming this serial record represents the journal containing *Hui*.

148. The serial record includes several ways for users to discover this journal, including the ISSN, CODEN, journal title, and two variant titles in Field 246. In addition, IEEE and the IEEE Communications Society are included in Fields 710 to help users discover this journal.

149. Interested users could also conduct subject searches to discover this journal, because the serial record contains subject representations of this journal, Field 050 shows a Library of Congress Classification number ("TK5101.A1") that

Google Exhibit No. 1063
Page 0073

represents periodicals on "Telecommunication" (source: Library of Congress Classification, Schedule T at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), Field 082 shows a Dewey Decimal Classification number ("621.38/05") that represents journals on "Electronics, communications engineering" (source: Dewey Decimal Classification Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf), and Field 650 shows a Library of Congress subject heading ("Telecommunication |v Periodicals") where "|v" indicates a form subheading to indicate this publication is a journal on telecommunication.

150. This MARC record (**Ex. 1019B**) shows that *IEEE Journal on Selected Areas in Communications* is a long-running journal that began publication in 1983. The MARC record makes this journal discoverable, and users can discover it by the journal's ISSN, CODEN, title, variant titles, and two organizations related to the journal. They can also conduct subject searches to discover this journal.

C. Usage Records

151. Actual usage of a publication is reflected by the papers that make reference to it. The citation history on Google Scholar shows *Hui* was cited 81 times. **Ex. 1019C** is a list of four citations from 1996 and 1997 that I compiled. It shows the earliest citation appeared in July 1996, supporting my opinion that *Hui* became

Google Exhibit No. 1063
Page 0074

publicly accessible by February 6, 1996.

D. Summary of My Opinion on *Hui*

152. The date stamp on **Ex. 1019A** shows the Library of Congress received the issue containing *Hui* on "JAN 30 1996." As discussed above, a customary library practice is to document the receipt of a new journal issue by placing the issue through the serial check-in system. After that, the new issue is promptly processed for public access, usually within a week after serial check-in is completed. It is therefore my opinion that the issue containing *Hui* was publicly accessible at the Library of Congress by February 6, 1996. Citation history shows the earliest citation was published in July 1996 (**Ex. 1019C**). The MARC record for the journal containing *Hui* was first created on August 2, 1982 (**Ex. 1019B**), and it shows interested users could discover this journal at the Library of Congress by conducting subject searches and searching by the journal's ISSN, CODEN, title, and variant titles.

XI. Public Availability of *Symes* (Ex. 1020)

153. **Ex. 1020** is a true and correct copy of excerpts of *Digital Video Compression*, ("*Symes*"), by Peter Symes, published by McGraw-Hill, I received from counsel. This copy is presented as **Ex. 1020** in this declaration.

154. Page 1 is the cover that shows the author and title, and indicates a CD is included *Symes*. Page 2 is the title page that shows the title, author, publisher and publisher locations. Page 3 is the copyright page that shows "Cataloging-in-

Google Exhibit No. 1063
Page 0075

Publication Data is on file with the Library of Congress," *Symes* has a "2004" copyright date, and the publisher holds the copyright. It also shows the book's ISBN (International Standard Book Number) is "0-07-142487-3" and indicates "[p]ortions of this book were previously published in *Video Compression Demystified*, ©2001." The table of contents appears from page 4 (internal page vii) to page 11 (internal page xiv) and shows *Symes* includes 22 chapters, a glossary, a bibliography that begins on internal page 377, a list of Internet Resources, a section on "About the CD-ROM," and an index.

A. Library of Congress Records

155. To determine the public availability date of *Symes*, I obtained Library of Congress records for this work. **Ex. 1020A** is a true and correct copy of the bibliographic and MARC records for *Symes*. I personally identified, located, and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

156. Pages 1 and 2 of **Ex. 1020A** are the bibliographic record that shows the author, title, publisher, and publication year of this work. It shows the ISBN ("0071424873"), subjects, and other details. The author, title, publisher, publication year ("c2004"), and ISBN match the information presented in **Ex. 1020**, confirming this record represents *Symes*. The "Item Availability" area shows the Library of

Google Exhibit No. 1063
Page 0076

Congress holds two copies of this book, with copy 1 stored at the Cabin Branch storage facility, and copy 2 stored at the Fort Meade storage facility. The call number of this book is "TK6680.5 .S93 2004," users can request access at the "Machine Readable Collections" of the Library of Congress, and the status of the copies is "Not Charged" (meaning available for user request).

157. Pages 3 and 4 are the MARC record for *Symes*. The first six digits of Field 008 show the MARC record was created on "030912" (*i.e.*, September 12, 2003), and the "s2004" code following the record creation date shows *Symes* has a single publication date of 2004. Field 955 records local cataloging activities and shows the library created a pre-publication record on "2003-09-12" (*i.e.*, September 12, 2003), received a copy of the published work on "2004-05-19" (*i.e.*, May 19, 2004) for "CIP ver." (*i.e.*, record verification at the Cataloging in Publication Program), completed the cataloging of copy 1 and sent it to BCCD (the Binding and Collection Care Division) for processing on "2004-06-15" (*i.e.*, June 15, 2004). Subfields "a" and "c" of Field 040 shows that "DLC" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library symbol for the Library of Congress. Fields 008, 955 and 040 inform my opinion that Library of Congress created an original MARC record for *Symes* on September 12, 2003, before the book was published, received the book for CIP record

verification on May 19, 2004, and sent copy 1 to the processing unit on June 15, 2004. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that *Symes* became publicly accessible at the Library of Congress no later than September 15, 2004.

158. Field 010 of the MARC record shows the Library of Congress Control Number and Field 020 show the ISBN of this book. Field 100 shows the author, Field 245 shows the title and statement of responsibility, and Field 260 shows the publisher and publication year. Field 300 shows the book's front matter ends on page xviii, the last numbered page is page 394, and the book includes illustrations and a CD-ROM. Field 500 shows this book is "Revision and extension of: Video compression demystified." Field 700 shows the author and the earlier title are included to help users discover this book.

159. Subjects of this book are represented in Field 050 by a Library of Congress Classification (LCC) number of "TK6680.5" that represents general works on "Digital video" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of "621.388" that

represents "Television" (source: Dewey Decimal Classification Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf). Subjects are also represented in Fields 650 by four Library of Congress subject headings: "Video compression," "Digital video," "MPEG (Video coding standard)," and "JPEG (Image coding standard)." These subject representations enable users to conduct subject searches to discover *Symes*.

160. This MARC record (**Ex. 1020A**) shows that the Library of Congress created a pre-publication record for *Symes* on September 12, 2003, completed their cataloging record for copy 1 on June 15, 2004, and sent the book to processing on the same day. These dates inform my opinion that *Symes* has become discoverable in the online catalog of the Library of Congress since June 15, 2004. As of that date, interested users could conduct subject searches by the LCC number, DDC number and the Library of Congress subject headings in the MARC record to discover this book. They could also search for this book by the author, title, and ISBN.

B. Summary of My Opinion on *Symes*

161. Based on the document and records I have reviewed and analyzed, it is my opinion that *Symes* has been publicly discoverable in the online catalog of the Library of Congress since June 15, 2004. The physical copy, after processing, would become publicly accessible at the Library of Congress by September 15, 2004.

XII. Public Availability of *Pereira* (Ex. 1021)

162. **Ex. 1021** is a true and correct copy of excerpts of *The MPEG-4 book*, ("*Pereira*"), by F. C. Pereira and T. Ebrahimi, published by Prentice Hall PTR, that I received from counsel. This copy is presented as **Ex. 1021** in this declaration.

163. Page 1 is the cover that shows the title, editors, and series title. Page 2 is the title page that shows the title, editors, publisher, and publisher's location and website URL. Page 3 is the copyright page that shows a record of "Library of Congress Cataloging-in-Publication Data," a "2002" copyright date with Pearson Education, Inc. as the copyright holder, and an ISBN (International Standard Book Number) of "0-13-061621-4." The table of contents from page 4 (internal page vii) and to page 15 (internal page xviii) shows that *Pereira* contains 16 chapters, five lettered sections, and an index. Page 92 is the spine that shows a call number, and page 93 is the back cover.

A. Library of Congress Records

164. To determine the public availability date of *Pereira*, I obtained Library of Congress records for this work. **Ex. 1021A** is a true and correct copy of the bibliographic and MARC records for *Pereira*. I personally identified, located, and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

Google Exhibit No. 1063
Page 0080

165. Pages 1 to 3 of **Ex. 1021A** are the bibliographic record that shows the first author, title, publisher, and publication year of this work. It shows the 10-digit ISBN ("0130616214"), second author, bibliographical and index note, series, and other details. The authors, title, publisher, publication year, and the 10-digit ISBN match the information presented in **Ex. 1021**, confirming this record represents *Pereira*. The "Item Availability" area shows the call number of *Pereira* is "TK6680.5 .P47 2002," copy 1 and copy 2 are stored at "CABIN BRANCH" (*i.e.*, the Cabin Branch storage facility of the Library of Congress), users can request access to these copies at the "Jefferson or Adams Building Reading Rooms" and the status of both copies is "Not Charged" (meaning available for user request).

166. Pages 3 and 4 are the MARC record for *Pereira*. The first six digits of Field 008 show the MARC record was created on "020509" (*i.e.*, May 9, 2002). Field 955 records local cataloging activities and shows a pre-publication record was created on "2002-05-09" (*i.e.*, May 9, 2002), the book was received on "2002-11-07" (*i.e.*, November 7, 2002) for "CIP ver." (*i.e.*, record verification at the Cataloging in Publication Program), and copy 1 was sent to BCCD (the Binding and Collection Care Division) for processing on "2002-11-12" (*i.e.*, November 12, 2002). Subfields "a" and "c" of Field 040 shows that "DLC" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library

symbol for the Library of Congress. Fields 008, 955 and 040 inform my opinion that Library of Congress created an original MARC record for *Pereira* on May 9, 2002, before the book was published, received the book for verification of the Cataloging in Publication record on November 7, 2002, completed the record and sent the book to the processing unit on November 12, 2002. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that *Pereira* became publicly accessible at the Library of Congress no later than February 12, 2003.

167. Field 010 of the MARC record shows the Library of Congress Control Number and Field 020 show the ISBN of this book. Field 100 shows the first author, Field 245 shows the title and statement of responsibility, and Field 260 shows the publisher and publication year. Field 490 shows the series title as presented in the book and Field 830 shows the series title in an authorized form for cataloging. Field 700 shows the second author is included to help users discover this book.

168. Subjects of this book are represented in Field 050 by a Library of Congress Classification (LCC) number of “tk6680.5” that represents general works on “Digital video” (source: Library of Congress Classification, Schedule T, at

https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of "621.388" that represents "Television" (source: Dewey Decimal Classification Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf). Subjects are also represented in Fields 650 by two Library of Congress subject headings, with "MPEG (Video coding standard)" and "Video compression |x Standards" ("|x" indicates a topical subheading) as the main topics. These subject representations enable users to conduct subject searches to discover *Pereira*.

169. This MARC record (**Ex. 1021A**) shows that the Library of Congress created a pre-publication record for *Pereira* on May 9, 2002, and completed their cataloging record and sent the book to processing on November 12, 2002. These dates inform my opinion that *Pereira* has been discoverable in the online catalog of the Library of Congress since November 12, 2002. As of that date, interested users could conduct subject searches by the LCC number, DDC number and the Library of Congress subject headings in the MARC record to discover *Pereira*. They could also search for this book by the authors, title, and ISBN.

B. Copyright Registration Record

170. **Ex. 1021B** is a true and correct copy of the copyright registration record for *Pereira*. I personally located, identified, and obtained this record from the public catalog of the U.S. Copyright Office. This is the type of record experts in my field

would reasonably rely upon when forming their opinions.

171. The title, authors, imprint, series, and copyright claimant, match the information presented in **Ex. 1021**, confirming this record represents *Pereira*. The Date of Publication field shows this book was published on "2002-07-10" (*i.e.*, July 10, 2002). These dates align well with the dates in the Library of Congress MARC record that show the library created a pre-publication record on May 9 2002, and received a copy of the book for verification of the Cataloging-in-Publication record on November 7, 2002.

C. Summary of My Opinion on *Pereira*

172. Based on the document and records I have reviewed and analyzed, it is my opinion that *Pereira* was published on July 10, 2002, and has been publicly discoverable in the online catalog of the Library of Congress since November 12, 2002. The physical copy, after processing, would become publicly accessible at the Library of Congress by February 12, 2003.

XIII. Public Availability of *Hunt* (Ex. 1022)

173. **Ex. 1022** is a true and correct copy of excerpts of *Windows NT TCP/IP Network Administration,* ("*Hunt*"), by Craig Hunt and Robert Bruce Thompson, published by O'Reilly, that I received from counsel. This copy is presented as **Ex. 1022** in this declaration.

174. Page 1 is the cover that shows the title, authors, and publisher. Page 2

Google Exhibit No. 1063
Page 0084

is the title page that shows the title, authors, publisher, and publisher's locations. Page 3 is the copyright page that shows a "1998" copyright date with the publisher as the copyright holder. This page shows that the version is "First Edition" published in October 1998. It also shows an ISBN (International Standard Book Number) of "1-56592-377-4." The table of contents from page 4 (internal page vii) and to page 7 (internal page x) shows that *Hunt* contains 13 chapters, four lettered sections, and an index.

A. Library of Congress Records

175. To determine the public availability date of *Hunt*, I obtained Library of Congress records for this work. **Ex. 1022A** is a true and correct copy of the bibliographic and MARC records for *Hunt*. I personally identified, located, and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

176. Pages 1 to 2 of **Ex. 1022A** are the bibliographic record that shows the first author, title, edition number, publisher, and publication year of this work. It shows the ISBN ("1565923774"), second author, and index note, and other details. The authors, title, publisher, publication year, and the ISBN match the information presented in **Ex. 1022**, confirming this record represents *Hunt*. The "Item Availability" area shows the call number of *Hunt* is "TK5105.585 .H864 1998," and

Google Exhibit No. 1063
Page 0085

users can request access to these copies at the "Jefferson or Adams Building Reading Rooms."

177. Pages 3 and 4 are the MARC record for *Hunt*. The first six digits of Field 008 show the MARC record was created on "010123" (*i.e.*, January 23, 2001), and the code of "s1998" following the record creation date indicates *Hunt* has a single publication date of 1998. Field 955 records local cataloging activities and shows the book was sent to ASCD (Arts and Sciences Cataloging Division) on "01-23-01" (*i.e.*, January 23, 2001), and the cataloging work was completed on "03-14-01" (*i.e.*, March 14, 2001). Subfields "a" and "c" of Field 040 shows that "CUS" created the original record and entered it into the OCLC WorldCat database, and subfield "d" shows "DLC" modified the original record later. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "CUS" is the OCLC library symbol for the University of California, San Diego, and "DLC" is the OCLC library symbol for the Library of Congress. Fields 008, 955 and 040 inform my opinion that CUS created an original MARC record for *Hunt* on January 23, 2001, and the Library of Congress used that record for copy cataloging later. Field 042 confirms that this record is "lccopycat" (meaning a copy cataloging record of the Library of Congress). A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of

Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that *Hunt* became publicly accessible at the Library of Congress no later than June 14, 2001.

178. Field 010 of the MARC record shows the Library of Congress Control Number and Field 020 show the ISBN of this book. Field 100 shows the first author, Field 245 shows the title and statement of responsibility, Field 246 shows a variant title, Field 250 shows the edition statement, and Field 260 shows the publisher and publication year. Field 700 shows the second author is included to help users discover this book.

179. Subjects of this book are represented in Field 050 by a Library of Congress Classification (LCC) number of "TK5105.585" that represents "TCP/IP" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of "005.7/13769" that represents "Data in computer systems" (source: Dewey Decimal Classification Class 000 at https://www.oclc.org/content/dam/oclc/webdewey/help/000.pdf). Subjects are also represented in Fields 650 by three Library of Congress subject headings, with "TCP/IP (Computer network protocol), "Microsoft Windows NT," and "Computer networks |x Management" ("|X" indicates a topical subheading) as the main topics. These subject representations enable users to conduct subject searches to discover

Hunt.

180. This MARC record (**Ex. 1022A**) shows that the library of the University of California, San Diego created a record for *Hunt* on January 23, 2001, and Library of Congress used that original record for copy cataloging and completed their copy cataloging record on March 14, 2001. As of that date, interested users could conduct subject searches by the LCC number, DDC number and the Library of Congress subject headings in the MARC record to discover *Hunt*. They could also search for this book by the authors, title, and ISBN.

B. Copyright Registration Record

181. **Ex. 1022B** is a true and correct copy of the copyright registration record for *Hunt*. I personally located, identified, and obtained this record from the public catalog of the U.S. Copyright Office. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

182. The title, authors, edition statement, imprint, series, and copyright claimant match the information presented in **Ex. 1022**, confirming this record represents *Hunt*. The Date of Publication field shows this book was published on "1998-10-13" (*i.e.*, October 13, 1998). These dates align well with the dates in the Library of Congress MARC record that show the library of the University of California, San Diego created the original MARC record on January 23, 2001, and the Library of Congress completed their copy cataloging record on March 14, 2001.

Google Exhibit No. 1063
Page 0088

C. Summary of My Opinion on *Hunt*

183. Based on the document and records I have reviewed and analyzed, it is my opinion that *Hunt* was published on October 13, 1998, the library of the University of California, San Diego created the original record for Hunt on January 23, 2001, and Library of Congress completed their copy cataloging record on March 14, 2001, making this work, publicly discoverable in the online catalog of the Library of Congress since that date. The physical copy, after processing, would become publicly accessible at the Library of Congress by June 14, 2001.

XIV. Public Availability of *Comer* (Ex. 1023)

184. **Ex. 1023** is a true and correct copy of excerpts of the fourth edition of *Internetworking with TCP/IP*, Volume 1: Principles, protocols, and architectures, ("*Comer*"), by Douglas E. Comer, published by Prentice Hall, that I received from counsel. This copy is presented as **Ex. 1023** in this declaration.

185. Page 1 is the cover that shows the title, title of volume 1, edition statement, and author. Page 2 is the title page that shows the same information as the cover, and includes the publisher's location. Page 3 is the copyright page that shows a record of "Library of Congress Cataloging-in-Publication Data," a "2000" copyright date with Prentice Hall as the copyright holder, and an ISBN (International Standard Book Number) of "0-13-018380-6." The table of contents from page 6 to 21 shows *Comer* contains 33 chapters, two appendices, a bibliography, and an index.

Google Exhibit No. 1063
Page 0089

A. Library of Congress Records

186. To determine the public availability date of *Comer*, I obtained Library of Congress records for this work. **Ex. 1023A** is a true and correct copy of the bibliographic and MARC records for *Comer*. I personally identified, located, and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

187. Pages 1 and 2 of **Ex. 1023A** are the bibliographic record that shows the author, title, edition statement, publisher, and publication year of this work. It shows the ISBN ("0130183806), title of volume 1 as "Internetworking with TCP/IP," "Principles, protocols, and architecture" as the secondary title, a bibliographical and index note, and other details. The author, title, edition statement, publisher, publication year, and the ISBN match the information presented in **Ex. 1023**, confirming this record represents *Comer*. The "Item Availability" area shows the call number of *Comer* is "TK5105.585 .C66 2000," volume 1 of copy 2 was "in Process" on "04-20-2000" (*i.e.*, April 20, 2000), users can request access to library copies at the "Jefferson or Adams Building Reading Rooms" and the status of both copies is "Not Charged" (meaning available for user request).

188. Pages 3 and 4 are the MARC record for *Comer*. The first six digits of Field 008 show the MARC record was created on "000228" (*i.e.*, February 28, 2000).

Google Exhibit No. 1063
Page 0090

Field 955 records local cataloging activities and shows a pre-publication record was created on "02-28-00" (*i.e.*, February 28, 2000), volume 1 was received on "04-20-00" (*i.e.*, April 20, 2000) for "CIP ver." (*i.e.*, record verification at the Cataloging in Publication Program), and copy 1 was sent to BCCD (the Binding and Collection Care Division) for processing on "08-31-00" (*i.e.*, August 31, 2000). Subfields "a" and "c" of Field 040 shows that "DLC" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library symbol for the Library of Congress. Fields 008, 955 and 040 inform my opinion that Library of Congress created an original MARC record for *Comer* on February 28, 2000, before the book was published, received the book for verification of the Cataloging in Publication record on April 20, 2000, completed the record and sent the book to the processing unit on August 31, 2000. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that *Comer* became publicly accessible at the Library of Congress no later than December 1, 2000.

189. Field 010 of the MARC record shows the Library of Congress Control

Google Exhibit No. 1063
Page 0091

Number and Field 020 show the ISBN of this book. Field 100 shows the author, Field 245 shows the title and statement of responsibility, Field 250 shows the edition statement, and Field 260 shows the publisher began publishing this book in 2000. Field 505 shows the title of volume 1.

190. Subjects of this book are represented in Field 050 by a Library of Congress Classification (LCC) number of "TK5105.585" that represents "TCP/IP" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of "005.2" that represents "Programming for specific types of computers, for specific operating systems, for specific user interfaces" (source: Dewey Decimal Classification Class 000 at https://www.oclc.org/content/dam/oclc/webdewey/help/000.pdf). Subjects are also represented in Fields 650 by three Library of Congress subject headings with "TCP/IP (Computer network protocol)" "Client/server computing," and "Internetworking (Telecommunication)" as the main topics. These subject representations enable users to conduct subject searches to discover *Comer*.

191. This MARC record (**Ex. 1023A**) shows that the Library of Congress created a pre-publication record for *Comer* on February 28, 2000, and completed their cataloging record and sent the book to processing on August 31, 2000. These dates inform my opinion that *Comer* has been discoverable in the online catalog of the

Library of Congress since August 31, 2000. As of that date, interested users could conduct subject searches by the LCC number, DDC number and the Library of Congress subject headings in the MARC record to discover *Comer*. They could also search for this book by the author, title, and ISBN.

B. Copyright Registration Record

192. **Ex. 1023B** is a true and correct copy of the copyright registration record for *Comer*. I personally located, identified, and obtained this record from the public catalog of the U.S. Copyright Office. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

193. The title, author, edition statement, imprint, and copyright claimant match the information presented in **Ex. 1023**, confirming this record represents *Comer*. The Date of Publication field shows this book was published on "2000-01-18" (*i.e.*, January 18, 2000). These dates align well and are consistent with the dates in the Library of Congress MARC records.

C. Summary of My Opinion on *Comer*

194. Based on the document and records I have reviewed and analyzed, it is my opinion that has been publicly discoverable in the online catalog of the Library of Congress since August 31, 2000. The physical copy, after processing, would become publicly accessible at the Library of Congress by December1, 2000.

Google Exhibit No. 1063
Page 0093

XV. Public Availability of *Stockhammer* (Ex. 1027)

195. **Ex. 1027** is a true and correct copy of "H.264/AVC in wireless environments," by Stockhammer et al., *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), 657-673, that I received from counsel. This journal issue has an issue date of "July 2003." This copy is presented as **Ex. 1027** in this declaration.

196. Page 1 (internal page 657) shows the title, authors, abstract, and index terms. The bottom of the page shows *Stockhammer* appears in a journal whose International Standard Serial Number (ISSN) is "1051-8215" and this article has a "2003" copyright date and IEEE (Institute of Electrical and Electronics Engineers, Inc.) holds the copyright. **Ex. 1027** shows *Stockhammer* has a total of 17 pages in six sections that includes 13 figures, two tables, three equations, and 70 references.

A. Date Stamped Copy from Library of Congress

197. To determine the public availability of *Stockhammer*, I searched WorldCat.org, the free web portal, to identify libraries holding the journal containing *Stockhammer*, and learned that the Library of Congress was one of the holding libraries. After verifying the holdings information in their online catalog, I requested a copy of *Stockhammer* from the Library of Congress through a document vendor. This library copy of *Stockhammer* is presented as **Ex. 1027A** in this declaration.

198. Page 1 of **Ex. 1027A** shows the spine of a bound volume identified as

Google Exhibit No. 1063
Page 0094

"13, 2003" of "IEEE Transactions on Circuits and Systems for Video Technology" that has a call number of "TK 6630 .A1 I37." Page 2 is the issue cover that shows this journal is a publication of the IEEE Circuits and Systems Society, its ISSN is 1051-8215, and its CODEN (a unique six- character code that identifies journals) is "ITCTEM." The cover also identifies this issue as volume 13, number 7 that has a "July 2003" issue date. The table of contents on this cover shows *Stockhammer* begins on page 657. Page 4 is the copyright page that shows this journal is published monthly by the Institute of Electrical and Electronics Engineers, Inc., and this issue has a "2003" copyright date with the publisher as the copyright holder. Page 5 is a page that carries a date stamp of "Library of Congress NOV 08 2003 Copyright Office" that indicates this issue was received by the Library of Congress on that date. Pages 6 to 22 (internal pages 657 to 673) are the full text of *Stockhammer*.

199. I have compared **Ex. 1027** with the Library of Congress copy of *Stockhammer* (**Ex. 1027A**) and found them to have the same content. The date stamp on **Ex. 1027A** indicates Library of Congress received this issue containing *Stockhammer* on November 8, 2003. As discussed above, a customary library practice is to document the receipt of a new journal issue by placing the issue through the serial check-in system. After that, the new issue is promptly processed for public access, usually within a week after serial check-in is completed. It is therefore my opinion that the issue containing *Stockhammer* became publicly accessible at the

Google Exhibit No. 1063
Page 0095

Library of Congress by November 15, 2003.

B. Library of Congress Records

200. As evidence that interested users could discover the journal holding *Stockhammer* at the Library of Congress to access this article, I obtained serial records for *IEEE Transactions on Circuits and Systems for Video Technology* from the online catalog of the Library of Congress. **Ex. 1027B** is a true and correct copy of the bibliographic and MARC records for this journal. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

201. Pages 1 to 4 of **Ex. 1027B** are the bibliographic record and Pages 5 to 8 are the MARC record for this journal. Field 008 of the MARC record shows the serial record was created on "900724" (*i.e.*, July 24, 1990), and the "c19919999" code following the record creation date indicates this publication is a continuing publication that began in 1991 and has no end date for publication yet. Subfields "a" and "c" of Field 040 show "NSD" (the National Serial Data Program at the Library of Congress) created the original record, and subfield "d" shows "DLC" (the Library of Congress) modified the record later. Field 022 shows the ISSN, Field 030 shows the CODEN, Field 245 shows the journal title, and Field 260 shows IEEE began publishing this journal in 1991. These details match the information presented in **Ex. 1027A**, confirming this serial record represents the journal containing *Stockhammer*.

Google Exhibit No. 1063
Page 0096

202. The serial record includes several ways for users to discover this journal, including the ISSN, CODEN, journal title, and two variant titles in Field 246. In addition, the IEEE Circuits and Systems Society is included in Field 710 to help users discover this journal.

203. Interested users could also conduct subject searches to discover this journal, because the serial record contains subject representations of this journal, Field 050 shows a Library of Congress Classification number ("TK6630.A1") that represents periodicals on "Television" (source: Library of Congress Classification, Schedule T at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), Field 082 shows a Dewey Decimal Classification number ("621") that represents "Applied physics" (source: Dewey Decimal Classification, Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf), and Fields 650 show four Library of Congress subject headings, with "Television |x Equipment and supplies" ("|x" indicates a topic subheading), "Television circuits," "Image processing," and "Signal processing" as the main topics, each followed by "Periodicals" as a form subheading encoded in subfield "v" to indicate the main topics are treated in a journal.

204. This MARC record (**Ex. 1027B**) shows that *IEEE Transactions on Circuits and Systems for Video Technology* is a long-running journal that began

Google Exhibit No. 1063
Page 0097

publication in 1991. The MARC record makes this journal discoverable, and users can discover it by the journal's ISSN, CODEN, title, variant titles, and the parent organization of this journal. They can also conduct subject searches to discover this journal.

C. Summary of My Opinion on *Stockhammer*

205. The date stamp on **Ex. 1027A** shows the Library of Congress received the issue containing *Stockhammer* on "November 08 2003." As discussed above, a customary library practice is to place a newly received journal issue through the serial check-in process to make the new issue publicly accessible as soon as possible, usually within a week after serial check-in is completed. It is therefore my opinion that the issue containing *Stockhammer* was publicly accessible at the Library of Congress by November 15, 2003. The MARC record for the journal containing *Stockhammer* was first created on July 24, 1990, and shows interested users could discover this journal at the Library of Congress by conducting subject searches and by searching by the journal's ISSN, CODEN, title, and variant titles.

XVI. Public Availability of *Richardson* (Ex. 1028)

206. **Ex. 1028** is a true and correct copy of excerpts of *H.264 and MPEG-4 Video Compression*, ("*Richardson*"), by I. E. G. Richardson, published by Wiley, that I received from counsel. This copy is presented as **Ex. 1028** in this declaration.

207. Page 1 is the cover that shows the author, title, and publisher. Page 3 is

Google Exhibit No. 1063
Page 0098

the title page that shows the same title, author, and publisher as the cover, and includes "video coding for next-generation multimedia" as the subtitle. Page 4 is the copyright page that shows a "2003" copyright date with the publisher as the copyright holder. It also includes "0-470-84837-5" as the book's ISBN (International Standard Book Number). The table of contents from page 6 to 10 shows *Richardson* includes eight chapters, a bibliography, and an index.

A. Library Records of University of Essex

208. To determine the public availability date of *Richardson*, I searched for this book in WorldCat.org, the free web portal, and learned that the library of the University of Essex created an original record for this book, so I obtained their library records. **Ex. 1028A** is a true and correct copy of the bibliographic and MARC records for *Richardson*. I personally identified, located, and obtained these records from the online catalog of the University of Essex library. They are the type of records experts in my field would reasonably rely upon when forming their opinions. I used screen captures to produce this exhibit.

209. Pages 1 to 3 of **Ex. 1028A** are the bibliographic record that shows *Richardson* is available at "Colchester, Albert Sloman Library Floor 2 north (TK6680.5.R5)." The record details show the title, creator, publisher, creation date, format, subjects, and identifier. The author, title, publisher, and publication year match the information presented in **Ex. 1028**, confirming this record represents

Google Exhibit No. 1063
Page 0099

Richardson.

210. Page 4 is the MARC record for *Richardson*. The first six digits of Field 008 show the MARC record was created on "031106" (*i.e.*, November 6, 2003), and the "s2003" code following the record creation date shows *Richardson* has a single publication date of 2003. Subfields "a" and "c" of Field 040 shows that "EUE" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "EUE" is the OCLC library symbol for the University of Essex library. Fields 008 and 040 inform my opinion that the library of the University of Essex created an original cataloging record for *Richardson* on November 6, 2003, making this book discoverable at their library from that date on. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed, if there is any delay, the processing is usually completed within a month. It is therefore my opinion that *Richardson* became publicly discoverable at the library of the University of Essex on November 6, 2003. The physical copy, after processing, would become publicly accessible by November 13, 2003, and definitely no later than December 6, 2003.

211. Field 100 shows the author, Field 245 shows the title, subtitle, and statement of responsibility, and Field 260 shows the publisher and the copyright date.

Google Exhibit No. 1063
Page 0100

212. Subjects of this book are represented in Field 099 by a Library of Congress Classification (LCC) number of "TK6680.5.R5" that represents general works on "Digital video" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf). Subjects are also represented in Fields 650 by three subject headings, with "Video compression," "MPEG (Video coding standard)," and "Multimedia systems" as the main topics. These subject representations enable users to conduct subject searches to discover *Richardson*.

213. This MARC record (**Ex. 1028A**) shows that the library of the University of Essex created an original cataloging record for *Richardson* on November 6, 2003, making this book discoverable from that date on. As of that date, interested users could conduct subject searches by the LCC number, and the subject headings in the MARC record to discover this book. They could also search for this book by the author and title.

B. Library Records of the Texas A&M University

214. My research on WorldCat also identified the Texas A&M University Libraries as a holding library of *Richardson*. **Ex. 1028B** is a true and correct copy of the bibliographic and MARC records for this book. I personally located, identified, and obtained the records from the online catalog of the Texas A&M University Libraries. They are the type of records experts in my field would reasonably rely upon

when forming their opinions.

215. Page 1 of **Ex. 1028B** is the bibliographic record that shows the title, author, format, language, publisher, and subjects of *Richardson*. This page includes the MARC record. The first six digits of Field 008 show the MARC record was created on "031124" (*i.e.*, November 24, 2003), and the "s2003" code following the record creation date indicates *Richardson* has a single publication date of 2003. Subfields "a" and "c" of Field 040 shows that "UKM" created the original record and entered it into the OCLC WorldCat database, and subfield "d" shows "TXA" modified the record later. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "UKM" is the OCLC library symbol for the British Library, and "TXA" is the OCLC library symbol for the Texas A&M University Libraries. Field 948 records local cataloging activities and shows *Richardson* was "cataloged" on "2004/1/28" (*i.e.*, January 28, 2004). Fields 008, 040, and 948 inform my opinion that the British Library created an original record for *Richardson* on November 24, 2003, and the Texas A&M University Libraries used it for copy cataloging and completed their copy cataloging record on January 28, 2004, making this book discoverable at their library from that date on. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed, if there is any delay, the processing is usually completed within a month. It is therefore my opinion that

Google Exhibit No. 1063
Page 0102

Richardson became publicly accessible at the library of the Texas A&M University libraries by February 4, 2004, and definitely no later than February 28, 2004.

216. Field 100 shows the author, Field 245 shows the title, subtitle, and statement of responsibility, and Field 263 shows the publisher and the copyright date.

217. Subjects of this book are represented in Field 090 by a Library of Congress Classification (LCC) number of "TK6680.5" that represents general works on "Digital video" (source: Library of Congress Classification, Schedule T, at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf). Subjects are also represented in Fields 650 by four Library of Congress subject headings for four main topics: "Digital video |x Standards" ("|x" indicates a topical subheading), "Video compression," "MPEG (Video coding standard)," and "Coding theory." These subject representations enable users to conduct subject searches to discover *Richardson*.

218. This MARC record (**Ex. 1028B**) shows that the Texas A&M University Libraries used an original record created by the British Library on November 24, 2003, to create their copy cataloging record and completed their record on January 28, 2004, making this book discoverable from that date on. As of that date, interested users could conduct subject searches by the LCC number, and the Library of Congress subject headings in the MARC record to discover this book. They could also search for this book by the author and title. The physical copy, after processing,

would become publicly accessible by February 28, 2004.

C. Summary of My Opinion on *Richardson*

219. Based on the document and records I have reviewed and analyzed, it is my opinion that *Richardson* has been publicly discoverable in the online catalog of the University of Essex since November 6, 2003. The physical copy, after processing, would become publicly accessible at that library by December 6, 2003. I also found that *Richardson* became discoverable at the Texas A&M University Libraries on January 28, 2004, and the physical copy, after processing, would become publicly accessible by February 28, 2004, at this library system.

XVII. Public Availability of *Bourge* (Ex. 1029)

220. **Exhibit 1029** is a true and correct copy of "Low-power H. 264 video decoder with graceful degradation," ("*Bourge*"), by A. Bourge and J. Jung in *Visual Communications and Image Processing 2004* (Proceedings of SPIE Vol. 5308), pp. 372-383, that I received from counsel. This copy is presented as **Ex. 1029** in this declaration.

221. Page 1 of (internal page 327) shows the title, authors, abstract, and keywords of *Bourge*. The bottom of the page shows this article is included in *Visual Communications and Image Processing 2004*, which is Proceedings of SPIE Vol. 5308, and the article has a "2004" copyright date with SPIE and IS&T as copyright holders. **Ex. 1029** shows *Bourge* has 12 pages in five sections, including ten figures,

Google Exhibit No. 1063
Page 0104

five tables, one equation, and 18 references.

A. Linda Hall Library Copy

222. To determine the public availability date of *Bourge*, I searched WorldCat.org, the free web portal, for libraries holding the conference proceedings containing *Bourge*, and learned that Linda Hall Library was one of the holding libraries. After verifying the holdings information in their online catalog, I obtained a copy of *Bourge* from this library through their document delivery service. This library copy is presented as **Ex. 1029A** in this declaration.

223. Pages 1 to 12 (internal pages 372 to 383) are the full text of *Bourge*. I have compared this copy with **Ex. 1029** and found them to contain the same content. Page 13 is the cover that shows the conference name, two editors, the date and location of the conference, IS&T (The Society for Imaging Science and Technology) and SPIE (The International Society for Optical Engineering), and "SPIE Vol. 1 5308, Part One of Two Parts). The top of the page indicates this work is the proceedings of "Electronic Imaging Science and Technology." Page 14 is the title page that contains the same information as the cover and clarifies that IS&T and SPIE and the IEEE Circuits and Systems Society sponsored the conference, and IS&T and SPIE published the proceedings.

224. Page 15 is the copyright page that shows a "2004" copyright date with SPIE and IS&T as the copyright holders. This page shows "0-8194-5211-4" as the

Google Exhibit No. 1063
Page 0105

ISBN of this conference proceedings and a hand-written call number of "TA1632 .V56 2004 v. 1." Page 16 (internal page iii) shows a date annotation of "2-20-04" that indicates the Linda Hall Library added this publication to their collections on February 20, 2004. The knowledge of this annotation practice is based on my personal knowledge and experience with the Linda Hall Library collections. The table of contents runs from page 16 to 27, and page 19 (internal page vi) shows *Bourge* is in Session 10 "Video Coding H264 II" and begins on page 372. The entry identifies this paper as "5308-38."

B. Linda Hall Library Records

225. As evidence that interested users could discover *Visual Communications and Image Processing 2004* (Proceedings of SPIE Vol. 5308) that contains *Bourge*, I obtained records for this work from the online catalog of the Linda Hall Library. **Ex. 1029B** is a true and correct copy of the bibliographic and MARC records for this work. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

226. Pages 1 to 4 of **Ex. 1029B** are the bibliographic record and Page 5 is the MARC record. The title, editors, sponsors and publishers, ISBN and series information match the information presented in **Ex. 1029** and **Ex. 1029A**, confirming this record represents *Visual Communications and Image Processing 2004* that

Google Exhibit No. 1063
Page 0106

contains *Bourge*. Field 008 of the MARC record shows the record was created on "040220" (*i.e.*, February 20, 2004). Field 005 is a machine-generated field that automatically record the last date when the library modified the MARC record, and it shows "20040220132000" (*i.e.*, February 20, 2004, at 1 p.m., 20 minutes) is the last record modification date. Subfields "a" and "c" of Field 040 show "LHL" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "LHL" is the OCLC library symbol for Linda Hall Library. Based on the data in Fields 008, 005, and 040, it is my opinion that the Linda Hall Library created the original record for the conference proceedings containing *Bourge* on February 20, 2004, making this work discoverable in their online catalog from that date on. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. It is therefore my opinion that *Visual Communications and Image Processing 2004* that contains *Bourge* became publicly discoverable on February 20, 2004. The physical copy, after processing, would become publicly accessible by February 27, 2004.

227. Field 020 shows the ISBN, Field 245 shows the title, subtitle, and statement of responsibility, Fields 246 show two variant titles are included, and Field 260 shows the publisher and copyright date. Field 490 shows the series information presented in the publication and Field 830 shows the authorized form of the series

title and numbering. The editors are presented in Fields 700, and three organizations are included in Fields 710. These fields show interested users could use these data elements to discover the conference proceedings containing *Bourge*.

228. Interested users could also conduct subject searches to discover this conference proceedings, because the MARC record contains subject representations. Field 090 shows a Library of Congress Classification number ("TA1632") that represents general works on "Optical data processing" (source: Library of Congress Classification Schedule T at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and Fields 650 show two Library of Congress subject headings, with "Image processing," and "Coding theory" as the main topics, each followed by "Congresses" as a form subheading, encoded in subfield "v" to indicate the main topics are treated at a conference.

229. This MARC record (**Ex. 1029B**) shows that the Linda Hall Library cataloged *Visual Communications and Image Processing 2004* on February 20, 2004, making this work discoverable. As of that date, interested users could discover this work by title, variant titles, series, editors, and three organizations. They could also conduct subject searches to discover this conference proceedings.

C. Summary of My Opinion on *Bourge*

230. The date annotation on **Ex. 1029A** shows Linda Hall Library added

Google Exhibit No. 1063
Page 0108

Visual Communications and Image Processing 2004 to their collections on February 20, 2004, and the MARC record (**Ex. 1029B**) shows the cataloging work was completed on February 20, 2004, making this work discoverable in their online catalog. The physical copy, after processing, would become publicly accessible by February 27, 2004.

XVIII. Public Availability of *Cen* (Ex. 1032)

231. **Exhibit 1032** is a true and correct copy of "Flow and congestion control for internet media streaming applications," ("*Cen*"), by Cen, Walpole and Pu, in *Multimedia Computing and Networking 1998* (Proceedings of SPIE Vol. 3310), pp. 250-264, that I received from counsel. This copy of *Cen* is presented as **Ex. 1032** in this declaration.

232. Page 1 is a cover sheet for *Cen* from PDXScholar, the Portland State University Institutional Repository. Page 2 (internal page 250) shows the title, authors, abstract, and keywords. The bottom of the page shows this article is included in SPIE Vol. 3310, and the series' ISSN is "0277-786X." **Ex. 1032** shows *Cen* has 15 pages in seven sections, including ten figures, two tables, one equation, and 17 references.

A. Linda Hall Library Copy

233. To determine the public availability date of *Cen*, I searched WorldCat.org, the free web portal, for libraries holding the conference proceedings

Google Exhibit No. 1063
Page 0109

containing *Cen*, and learned that Linda Hall Library was one of the holding libraries. After verifying the holdings information in their online catalog, I obtained a copy of *Cen* from this library through their document delivery service. This library copy is presented as **Ex. 1032A** in this declaration.

234. Pages 1 to 15 (internal pages 250 to 264) are the full text of *Cen*. I have compared this copy with **Ex. 1032** and found them to contain the same content. Page 16 is the cover that shows the conference name, three editors, the date and location of the conference, SPIE (The International Society for Optical Engineering) and IS&T (The Society for Imaging Science and Technology) as the sponsors, and ACM SIGMultimedia as the cosponsor. The top of the page indicates this work is "Proceedings of SPIE" and SPIE is "The International Society for Optical Engineering." The lower right corner identifies this work as volume 3310. Page 17 is the title page that contains the same information as the cover and identifies SPIE as the publisher.

235. Page 18 is the copyright page that shows a "1997" copyright date with SPIE as the copyright holder. This page shows "0-8194-2750-0" as the ISBN of this conference proceedings and a hand-written call number of "TK5105.5 .M84 1998." Page 19 (internal page iii) shows a date annotation of "1-8-98" that indicates the Linda Hall Library added this publication to their collections on January 8, 1998. The knowledge of this annotation practice is based on my personal knowledge and

Google Exhibit No. 1063
Page 0110

experience with the Linda Hall Library collections. The table of contents runs from page 19 to 20, and page 20 (internal page iv) shows *Cen* is in Session 7 "Flow and Congestion Control" and begins on page 250. The entry identifies this paper as "3310-19."

B. Linda Hall Library Records

236. As evidence that interested users could discover *Multimedia Computing and Networking 1998* (Proceedings of SPIE Vol. 3310) that contains *Cen*, I obtained records for this work from the online catalog of the Linda Hall Library. **Ex. 1032B** is a true and correct copy of the bibliographic and MARC records for this work. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

237. Pages 1 to 4 of **Ex. 1032B** are the bibliographic record and Page 5 is the MARC record. The title, editors, sponsors, publisher, ISBN and series information match the information presented in **Ex. 1032** and **Ex. 1032A**, confirming this record represents *Multimedia Computing and Networking 1998* that contains *Cen*. Field 008 of the MARC record shows the record was created on "980109" (*i.e.*, January 9, 1998). Field 005 is a machine-generated field that automatically record the last date when the library modified the MARC record, and it shows "19980109083600" (*i.e.*, January 9, 1998, at 8 a.m., 36 minutes) is the last record modification date. Subfields "a" and "c" of Field 040 show "LHL" created the

Google Exhibit No. 1063
Page 0111

original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "LHL" is the OCLC library symbol for the Linda Hall Library. Based on the data in Fields 008, 005, and 040, it is my opinion that the Linda Hall Library created the original record for the conference proceedings containing *Cen* on January 9, 1998, making this work discoverable in their online catalog from that date on. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. It is therefore my opinion that *Multimedia Computing and Networking 1998* that contains *Cen* became publicly discoverable on January 9, 1998. The physical copy, after processing, would become publicly accessible by January 16, 1998.

238. Fields 020 show the ISBNs, Field 245 shows the title, subtitle, and statement of responsibility, and Field 260 shows the publisher and copyright date. Field 490 shows the series information presented in the publication and Field 830 shows the authorized form of the series title and numbering. The editors are presented in Fields 700, and three sponsors are included in Fields 710. These fields show interested users could use these data elements to discover the conference proceedings containing *Cen*.

239. Interested users could also conduct subject searches to discover this conference proceedings, because the MARC record contains subject representations.

Google Exhibit No. 1063
Page 0112

Field 090 shows a Library of Congress Classification number ("TK5105.5") that represents general works on "Computer networks" (source: Library of Congress Classification Schedule T at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), and Fields 650 show two Library of Congress subject headings, with "Multimedia systems" and "Computer networks" as the main topics, each followed by "Congresses" as a form subheading, encoded in subfield "v" to indicate the main topics are treated at a conference.

240. This MARC record (**Ex. 1032B**) shows that Linda Hall Library cataloged *Multimedia Computing and Networking 1998* on January 9, 1998, making this work discoverable. As of that date, interested users could discover this work by the ISBN, title, series, editors, and three sponsors. They could also conduct subject searches to discover this conference proceedings.

C. Summary of My Opinion on Cen

241. The date annotation on **Ex. 1032A** shows Linda Hall Library added *Multimedia Computing and Networking 1998* to their collections on January 8, 1998, and the MARC record (**Ex. 1032B**) shows the cataloging work was completed on January 9, 1998, making this work discoverable in their online catalog from that date on. The physical copy, after processing, would become publicly accessible by January 16, 1998.

Google Exhibit No. 1063
Page 0113

XIX. Public Availability of *Lu* (Ex. 1033)

242. **Exhibit 1033** is a true and correct copy of "Signal processing for Internet video streaming: A review," ("*Lu*"), by J. Lu in *Image and Video Communications and Processing 2000* (Proceedings of SPIE Vol. 3974), 14 unnumbered pages, that I received from counsel. This copy is presented as **Ex. 1033** in this declaration.

243. Page 1 shows the title, author, abstract, and keywords. The top of the page shows this article is included in "Proceedings of SPIE Image and Video Communications and Processing, January 2000." **Ex. 1033** shows *Lu* has a total of 14 pages, including three figures and 58 references.

A. Linda Hall Library Copy

244. To determine the public availability date of *Lu*, I searched WorldCat.org, the free web portal, for libraries holding the conference proceedings containing *Lu*, and learned that Linda Hall Library was one of the holding libraries. After verifying the holdings information in their online catalog, I obtained a copy of *Lu* from this library through their document delivery service. This library copy is presented as **Ex. 1033A** in this declaration.

245. Pages 1 to 14 (internal pages 246 to 259) are the full text of *Lu*. I have compared this copy with **Ex. 1033** and found them to contain the same content. One difference is that the pages in the library copy are numbered, and page 246 shows *Lu*

Google Exhibit No. 1063
Page 0114

is in "Image and Video Communications and Processing 2000," which is "Proceedings of SPIE Vol. 3974 (2000)," and the proceedings has four editors.

246. Page 15 is the cover that shows the conference name, four editors, the date and location of the conference, and IS&T (The Society for Imaging Science and Technology) and SPIE (The International Society for Optical Engineering) as the sponsors. The top of the page indicates this work is "Proceedings of SPIE, SPIE—The International Society for Optical Engineering" and lower right corner indicates this work is volume 3974.

247. Page 16 is the title page that shows the same information as the cover and clarifies that IS&T and SPIE are the sponsors, and SPIE is the publisher of this work. Page 17 is the copyright page that shows a "2000" copyright date with SPIE as the copyright holder. This page shows "0-8194-3592-9" as the ISBN of this conference proceedings and a hand-written call number of "TA1632 .I41123 2000." Page 18 (internal page iii) shows a date annotation of "5-5-00" that indicates the Linda Hall Library added this publication to their collections on May 5, 2000. The knowledge of this annotation practice is based on my personal knowledge and experience with the Linda Hall Library collections. The table of contents runs from page 18 to 25, and page 19 (internal page iv) shows *Lu* is in Session 7 "Image and Video Transmission I" and begins on page 246. The entry identifies this paper as "3974-26."

Google Exhibit No. 1063
Page 0115

B. Linda Hall Library Records

248. As evidence that interested users could discover *Image and Video Communications and Processing 2000* (Proceedings of SPIE Vol. 3974) that contains *Lu*, I obtained records for this work from the online catalog of the Linda Hall Library. **Ex. 1033B** is a true and correct copy of the bibliographic and MARC records for this work. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

249. Pages 1 to 3 of **Ex. 1033B** are the bibliographic record and Page 4 is the MARC record. The title, first editor, sponsors, publishers, ISBN and series information match the information presented in **Ex. 1033A**, confirming this record represents *Image and Video Communications and Processing 2000* that contains *Lu*. Field 008 of the MARC record shows the record was created on "000505" (*i.e.*, May 5, 2000). Subfields "a" and "c" of Field 040 show "LHL" created the original record and entered it into the OCLC WorldCat database. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "LHL" is the OCLC library symbol for Linda Hall Library. Based on the data in Fields 008 and 040, it is my opinion that Linda Hall Library created the original record for the conference proceedings containing *Lu* on May 5, 2000, making this work discoverable in their online catalog from that date on. A customary library practice is

Google Exhibit No. 1063
Page 0116

to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. It is therefore my opinion that *Image and Video Communications and Processing 2000* that contains *Lu* became publicly discoverable on May 5, 2000. The physical copy, after processing, would become publicly accessible by May 12, 2000.

250. Fields 020 show the ISBNs, Field 245 shows the title, subtitle, and statement of responsibility, and Field 260 shows the publisher and copyright date. Field 490 shows the series information presented in the publication and Field 830 shows the authorized form of the series title and numbering. The first editor is included in Fields 700, and the sponsors are included in Fields 710. These fields show interested users could use these data elements to discover the conference proceedings containing *Lu*.

251. Interested users could also conduct subject searches to discover this conference proceedings, because the MARC record contains subject representations. Field 090 shows a Library of Congress Classification number ("TA1632") that represents general works on "Optical data processing," and Fields 650 show four Library of Congress subject headings, with "Computer graphics," "Video compression," "Image compression," and "Image processing" as the main topics, each followed by "Congresses" as a form subheading, encoded in subfield "v", to indicate the main topics are treated at a conference.

Google Exhibit No. 1063
Page 0117

252. This MARC record (**Ex. 1033B**) shows that the Linda Hall Library cataloged *Image and Video Communications and Processing 2000* on May 5, 2000, making this work discoverable. As of that date, interested users could discover this work by ISBN, title, series, editor, and two sponsors. They could also conduct subject searches to discover this conference proceedings.

C. Summary of My Opinion on *Lu*

253. The date annotation on **Ex. 1033A** shows the Linda Hall Library added *Image and Video Communications and Processing 2000* to their collections on May 5, 2000, and the MARC record (**Ex. 1033B**) shows the cataloging work was completed on May 5, 2000, making this work discoverable in their online catalog. The physical copy, after processing, would become publicly accessible by May 12, 2000.

XX. Public Availability of *Wetzel* (Ex. 1038)

254. **Ex. 1038** is a true and correct copy of "Consumer Applications of the IEEE 1394 Serial Bus, and A 1394/DV video Editing System," ("*Wetzel*"), by A. T. Wetzel and M. R. Schell, that I received from counsel. This copy is presented as **Ex. 1038** in this declaration.

255. Page 1 is a cover sheet that shows *Wetzel* has a "1996" copyright date and the Institute of Electrical and Electronic Engineers, Inc. holds the copyright. The cover sheet indicates this copy is a reprint by permission of the IEEE, and the paper

Google Exhibit No. 1063
Page 0118

"was prepared for and presented at the International Conference on Consumer Electronics held in Chicago, Ill., June 1996, conference session WPM-6, Intelligent Home Networks." Pages 2 to 12 show the full text of *Wetzel* that has 11 unnumbered pages and includes an abstract, four figures, and 16 references.

A. Linda Hall Library Copy

256. To determine the public availability date of *Wetzel*, I searched Google Scholar for records of this article, and one record linked to the full text of *Wetzel* at https://www.ardent-tool.com/CPU/docs/AMD/anatomy/misc/articles/wetzel.pdf. The top of the cover sheet shows a note that the article is from the companion CD-ROM to the IEEE CS Press book, "The Anatomy of a Microprocessor: A Systems Perspective," by Shriver & Smith. I then searched for libraries that held this book and CD-ROM, and identified the Linda Hall Library and the Library of Congress as two of the holding libraries. **Ex. 1038A** is a copy of *Wetzel* and **Ex. 1038B** is a copy of the front matter of the book by Shriver & Smith book that I obtained from the Linda Hall Library. I have compared **Ex. 1038** with **Ex. 1038A** and found them to have the same content.

257. Page 1 of **Ex. 1038B** is the title page that shows "The Anatomy of a High-Performance Microprocessor: A Systems Perspective" as the title, Bruce Shriver and Bennett Smith as the authors, and IEEE Computer Society Press of Los Alamitos, California as the publisher. Page 2 is the copyright page that shows this

book has a "1998" copyright date and the Institute of Electrical and Electronics Engineers, Inc. is the copyright holder. This page shows a record of "Library of Congress Cataloging-in-Publication Data." It also shows "0-8186-8400-3" as the book's ISBN (International Standard Book Number), "98-17345" as the Library of Congress control number, and "BP08400" as the IEEE Computer Society Press order number. **Ex. 1038A** and **Ex. 1038B** inform my opinion that the book and the CD-ROM containing *Wetzel* are available for public access at the Linda Hall Library.

B. Linda Hall Library Records

258. As evidence that the book and CD-ROM containing *Wetzel* are available and discoverable at the Linda Hall Library, I obtained the library records for this book and its accompanying CD-ROM. I personally located, identified, and obtained the records from the online catalog of the Linda Hall Library. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

259. **Ex. 1038C** is a true and correct copy of the bibliographic and MARC records. Pages 1 to 3 are the bibliographic record that shows this book is available "at Linda Hall Library Books – 2nd Floor (QA76.5 .S49 1998) and other locations." The record shows the title, creators, subjects, ISBNs, publisher and publication year, physical description, and a general note. The title, authors, ISBNs, and publisher information match the information presented in **Ex. 1038B**, confirming this record

Google Exhibit No. 1063
Page 0120

represents the book whose accompanying CD-ROM contains *Wetzel*.

260. Pages 4 is a copy of the MARC record, and the first six digits of Field 008 show the record was first created on "980311" (*i.e.*, March 11, 1998). Subfields "a" and "c" of Field 040 show "DLC" created the original record and entered it into the OCLC WorldCat database, and subfield "d" shows "LHL" modified the record later. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library symbol for the Library of Congress, and "LHL" is the OCLC library symbol for the Linda Hall Library. Fields 008 and 040 inform my opinion that the Library of Congress created an original record for this book on March 11, 1998, and the Linda Hall Library used the original record to create their copy cataloging record later.

261. **Ex. 1038C** shows interested users could discover this book and its accompanying CD-ROM by the authors and title. In addition, subject representations in the MARC record enable users to conduct subject searches to discover this work. Field 050 shows a Library of Congress Classification number of "QA76.5" that represents general works on "Digital computers" (source: Library of Congress Classification Schedule Q at https://www.loc.gov/aba/publications/FreeLCC/LCC_Q2020TEXT.pdf), and Fields 650 show three Library of Congress subject headings are assigned to indicate this work covers "Microprocessors," "High performance computing," and "Computer

Google Exhibit No. 1063
Page 0121

architecture."

262. Since the Linda Hall Library MARC record indicates their copy cataloging record is based on a MARC record created by the Library of Congress, I obtained the bibliographic and MARC records from the Library of Congress to determine the earliest public availability date. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

263. **Ex. 1038D** is a true and correct copy of the bibliographic and MARC records. Pages 1 and 2 are the bibliographic record. Pages 3 and 4 are the MARC record. The first six digits of Field 008 of the MARC record show the record was first created on "980311" (*i.e.*, March 11, 1998). Field 955 is a field for local cataloging activities and it shows that after the original record was created, the verification of the Cataloging-in-Publication record took place on "01-18-00" (*i.e.*, January 18, 2000), and the publication was sent to BCCD (Binding and Collection Care Division) on "01-20-00" (*i.e.*, January 20, 2000) and "05-30-00" (*i.e.*, May 30, 2000). Subfields "a" and "c" of Field 040 show "DLC" (*i.e.*, the Library of Congress) created the original record and entered it into OCLC's WorldCat database. Fields 008, 955, and 040 inform my opinion that the MARC record was verified on January 18, 2000, and the publication was sent to processing on January 20, 2000, making this publication discoverable in the library's online catalog from that date on. As of that date,

Google Exhibit No. 1063
Page 0122

interested users could discover this work by the Library of Congress control number in Field 010, the ISBN in Field 020, the authors in Fields 100 and 700, and the title in Field 245. Interested users could also conduct subject searches to discover this work, because the record includes subject representations. Field 050 shows a Library of Congress Classification number of "QA76.5" that represents general works on "Digital computers" (source: Library of Congress Classification Schedule Q at https://www.loc.gov/aba/publications/FreeLCC/LCC_Q2020TEXT.pdf), Field 082 shows a Dewey Decimal Classification number of "004.16" that represents "Personal computers" (source: Dewey Decimal Classification Class 000 at https://www.oclc.org/content/dam/oclc/webdewey/help/000.pdf), and Fields 650 show three Library of Congress subject headings are assigned to indicate this work covers "Microprocessors," "High performance computing," and "Computer architecture."

264. The MARC record shows the publication containing *Wetzel* was sent to the processing unit at the Library of Congress on August 30, 2000. A customary library practice is to process a newly cataloged monograph promptly for public access, usually within a week after the cataloging work is completed. My personal experience with the collections at the Library of Congress is that it can take one to three months for a newly cataloged monograph to become publicly accessible. It is therefore my opinion that the book with its accompanying CD-ROM that contains

Wetzel became publicly accessible at the Library of Congress no later than August 30, 2000.

C. Copyright Registration Record

265. **Ex. 1038E** is a true and correct copy of the copyright registration record for the book and CD-ROM that contains *Wetzel*. The title, imprint, CD-ROM note, and copyright claimant match the information presented in the front matter of this work (**Ex. 1038B**). The "Date of Publication" field shows this work was published on "1998-07-01" (*i.e.*, July 1, 1998), and the "Registration Number / Date" field shows this work was registered with the US Copyright Office on "1998-10-15" (*i.e.*, October 15, 1998). These dates align well with the fact that the Library of Congress created an original record on March 11, 1998, before the book was published, and completed their record on January 20, 2000.

D. Summary of My Opinion on *Wetzel*

266. **Ex. 1038A** and **Ex. 1038B** show the publication containing Wetzel are available at the Linda Hall Library, and **Ex. 1038C** shows the Linda Hall Library used a record created by the Library of Congress for copy cataloging. Interested users could conduct subject searches to discover this publication. They could also use the Library of Congress control number, ISBN, authors, and title to discover this work. The MARC record shows the Library of Congress created a pre-publication record for this work on March 11, 1998, and completed their record on January 20, 2000,

Google Exhibit No. 1063
Page 0124

when the work was sent to the processing unit (**Ex. 1038D**). The copyright registration record shows the publication that contains *Wetzel* was published on July 1, 1998 (**Ex. 1038E**).

267. Based on the documents and records I have reviewed, it is my opinion that the work containing *Wetzel* has been publicly discoverable in the online catalog of the Library of Congress since January 20, 2000 when the record was completed and the publication was sent to the processing unit. The physical copy, after processing, would become publicly accessible at the Library of Congress by August 30, 2000.

XXI. Public Availability of *Paik* (Ex. 1039)

268. **Ex. 1039** is a true and correct copy of "Design of a novel synthesis filter for real-time MPEG-2 audio decoder implementation on a DSP chip," ("*Paik*"), by Won-Kyu Paik and Sun-Young Hwang, *IEEE Transactions on Consumer Electronics*, 45(4), 1119-1129, that I received from counsel. This journal issue has an issue date of "November 1999." This copy is presented as **Ex. 1039** in this declaration.

269. Page 1 (internal page 1119) shows the title, authors, and abstract. The bottom of the page shows *Paik* appears in a journal whose International Standard Serial Number (ISSN) is "0099-3063" and this article has a "1999" copyright date and IEEE (Institute of Electrical and Electronics Engineers, Inc.) holds the copyright.

Google Exhibit No. 1063
Page 0125

Ex. 1039 shows *Paik* has a total of 11 pages in five sections that includes four figures, four tables, four equations, and nine references.

A. Date Stamped Copy from Library of Congress

270. To determine the public availability of *Paik*, I searched WorldCat.org, the free web portal, to identify libraries holding the journal containing *Paik*, and learned that the Library of Congress was one of the holding libraries. After verifying the holdings information in their online catalog, I requested a copy of *Paik* from the Library of Congress through a document vendor. This library copy of *Paik* is presented as **Ex. 1039A** in this declaration.

271. Page 1 of **Ex. 1039A** shows the spine of a bound volume identified as "45, AUG-NOV 1999" of "IEEE Transactions on Consumer Electronics" that has a call number of "TK 6563 .I2." Page 2 is the issue cover that shows this journal is a publication of the IEEE Consumer Electronics Society, its ISSN is 0098-3063 and its CODEN (a unique six- character code that identifies journals) is "ITCEDA." The cover also identifies this issue as volume 45, number 4 that has a "November 1999" issue date. The table of contents on this page shows *Paik* begins on page 1119. Page 3 is the copyright page that shows this journal is published quarterly by the Institute of Electrical and Electronics Engineers, Inc., and this issue has a "1999" copyright date with the publisher as the copyright holder. Page 4 shows a date stamp of "Library of Congress JAN 10 2000 Copyright Office" that indicates this issue was received by

Google Exhibit No. 1063
Page 0126

the Library of Congress on that date. Pages 5 to 15 (internal pages 1119 to 1129) are the full text of *Paik*.

272. I have compared **Ex. 1039** with the Library of Congress copy of *Paik* (**Ex. 1039A**) and found them to have the same content. The date stamp on **Ex. 1039A** indicates Library of Congress received the issue containing *Paik* on January 10, 2000. As discussed above, a customary library practice is to document the receipt of a new journal issue by placing the issue through the serial check-in system. After that, the new issue is promptly processed for public access, usually within a week after serial check-in is completed. It is therefore my opinion that the issue containing *Paik* became publicly accessible at the Library of Congress by January 17, 2000.

B. Library of Congress Records

273. As evidence that interested users could discover the journal containing *Paik* at the Library of Congress to access this article, I obtained serial records for *IEEE Transactions on Consumer Electronics* from the online catalog of the Library of Congress. **Ex. 1039B** is a true and correct copy of the bibliographic and MARC records for this journal. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

274. Pages 1 to 4 of **Ex. 1039B** are the bibliographic record and Pages 5 to 8 are the MARC record for this journal. Field 008 of the MARC record shows the

Google Exhibit No. 1063
Page 0127

serial record was created on "751021" (*i.e.*, October 21, 1975), and the code of "c19759999" following the record creation date indicates this publication is a continuing publication that began in 1975 and has no end date for publication yet. Subfields "a" and "c" of Field 040 show "DLC" (the Library of Congress) created the original record. Field 022 shows the ISSN, Field 030 shows the CODEN, Field 245 shows the journal title, and Field 260 shows IEEE began publishing this journal in 1975. These details match the information presented in **Ex. 1039A**, confirming this serial record represents the journal containing *Paik*.

275. The serial record includes several ways for users to discover this journal, including the ISSN, CODEN, journal title, and a variant title in Field 246. In addition, three organizations related to this journal are included in Fields 710 and a conference is included in Field 711.

276. Interested users could also conduct subject searches to discover this journal, because the serial record contains subject representations of this journal, Field 050 shows a Library of Congress Classification number ("TK6563") that represents general works on "Radio receiving apparatus" (source: Library of Congress Classification Schedule T at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), Field 082 shows a Dewey Decimal Classification number ("621.38") that represents "Electronics, communications engineering" (source: Dewey Decimal Classification

Google Exhibit No. 1063
Page 0128

Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf), and Fields 650 show four Library of Congress subject headings, with "Household electronics," "Radio |x Receivers and reception" ("|x" indicates a topical subheading), "Television |x Receivers and reception," and "Electronic apparatus and appliances" as the main topics, each followed by "Periodicals" as a form subheading, encoded in subfield "v", to indicate the main topics are treated in a journal.

277. This MARC record (**Ex. 1039B**) shows that *IEEE Transactions on Consumer Electronics* is a long-running journal that began publication in 1975. The MARC record makes this journal discoverable, and users can discover it by the journal's ISSN, CODEN, title, variant title, three organizations and a conference name. They can also conduct subject searches to discover this journal.

C. Copyright Registration Record

278. **Ex. 1039C** is a true and correct copy of the copyright registration record for *IEEE Transactions on Consumer Electronics*. I personally located, identified, and obtained this record from the public catalog of the US Copyright Office. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

279. The title, serial publication year, imprint, ISSN, and copyright claimant match the information presented in **Ex. 1039A**, confirming this record represents *IEEE Transactions on Consumer Electronics*. The "Issues Registered" field shows v.

Google Exhibit No. 1063
Page 0129

45, no. 4, which contains *Paik*, has a publication date of "1999-12-17" (*i.e.*, December 17, 1999). This date aligns well with the date stamp on **Ex. 1039A** that shows the Library of Congress received v. 45, no. 4 on January 10, 2000.

D. Summary of My Opinion on *Paik*

280. The copyright registration record shows v. 45, no. 4 of *IEEE Transactions on Consumer Electronics* was published on December 17, 1999 (**Ex. 1039C**). The date stamp on **Ex. 1039A** shows the Library of Congress received the issue containing *Paik* on "JAN 10 2000." As discussed above, a newly received journal issue usually becomes publicly accessible within a week after serial check-in is completed. It is therefore my opinion that the issue containing *Paik* was publicly accessible at the Library of Congress by January 17, 2000. The MARC record for the journal containing *Paik* was first created on October 21, 1975 (**Ex. 1039B**), and shows interested users could discover this journal at the Library of Congress by conducting subject searches and by searching by the journal's ISSN, CODEN, title, and variant title.

XXII. Public Availability of *Lee* (Ex. 1040)

281. **Ex. 1040** is a true and correct copy of "Software optimization of the MPEG-audio decoder using a 32-bit MC RISC processor," ("*Lee*"), by Lee, Park and Youn, *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, 671-676, that I received from counsel. This journal issue has an issue date of "August 2002." This

Google Exhibit No. 1063
Page 0130

copy is presented as **Ex. 1040** in this declaration.

282. Page 1 (internal page 671) shows the title, authors, and abstract. The bottom of the page shows *Lee* appears in a journal whose International Standard Serial Number (ISSN) is "0098-3063" and this article has a "2002" copyright date and IEEE (Institute of Electrical and Electronics Engineers, Inc.) holds the copyright. **Ex. 1040** shows *Lee* has a total of six pages in five sections that includes five figures, one table, and six references.

A. Date Stamped Copy from Library of Congress

283. To determine the public availability of *Lee*, I searched WorldCat.org, the free web portal, to identify libraries holding the journal containing *Lee*, and learned that the Library of Congress was one of the holding libraries. After verifying the holdings information in their online catalog, I requested a copy of *Lee* from the Library of Congress through a document vendor. This library copy of *Lee* is presented as **Ex. 1040A** in this declaration.

284. Page 1 of **Ex. 1040A** shows the spine of a bound volume identified as "48, 2002" of "IEEE Transactions on Consumer Electronics" that has a call number of "TK 6563 .I2." Page 2 is the issue cover that shows this journal is a publication of the IEEE Consumer Electronics Society, its ISSN is 0098-3063 and its CODEN (a unique six-character code that identifies journals) is "ITCEDA." The cover also identifies this issue as volume 48, number 3 that has a "August 2002" issue date. The

Google Exhibit No. 1063
Page 0131

front cover and back cover show the table of contents, and the back cover on page 3 shows *Lee* begins on page 671. Page 4 is the copyright page that shows this journal is published quarterly by the Institute of Electrical and Electronics Engineers, Inc., and this issue has a "2002" copyright date with the publisher as the copyright holder. Page 5 shows a date stamp of "Copyright Office OCT 31 2002 Library of Congress" that indicates this issue was received by the Library of Congress on that date. Pages 6 to 11 (internal pages 671 to 676) are the full text of *Lee*.

285. I have compared **Ex. 1040** with the Library of Congress copy of *Lee* (**Ex. 1040A**) and found them to have the same content. The date stamp on **Ex. 1040A** indicates Library of Congress received this issue containing *Lee* on October 31, 2002. As discussed above, a customary library practice is to document the receipt of a new journal issue by placing the issue through the serial check-in system. After that, the new issue is promptly processed for public access, usually within a week after serial check-in is completed. It is therefore my opinion that the issue containing *Lee* became publicly accessible at the Library of Congress by November 7, 2002.

B. Library of Congress Records

286. As evidence that interested users could discover the journal containing *Lee* at the Library of Congress to access this article, I obtained serial records for *IEEE Transactions on Consumer Electronics* from the online catalog of the Library of Congress. **Ex. 1040B** is a true and correct copy of the bibliographic and MARC

Google Exhibit No. 1063
Page 0132

records for this journal. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

287. Pages 1 to 4 of **Ex. 1040B** are the bibliographic record and Pages 5 to 8 are the MARC record for this journal. Field 008 of the MARC record shows the serial record was created on "751021" (*i.e.*, October 21, 1975), and the code of "c19759999" following the record creation date indicates this publication is a continuing publication that began in 1975 and has no end date for publication yet. Subfields "a" and "c" of Field 040 show "DLC" (the Library of Congress) created the original record. Field 022 shows the ISSN, Field 030 shows the CODEN, Field 245 shows the journal title, and Field 260 shows IEEE began publishing this journal in 1975. These details match the information presented in **Ex. 1040A**, confirming this serial record represents the journal containing *Lee*.

288. The serial record includes several ways for users to discover this journal, including the ISSN, CODEN, journal title, and a variant title in Field 246. In addition, three organizations related to this journal are included in Fields 710 and a conference is included in Field 711 to help users discover this journal.

289. Interested users could also conduct subject searches to discover this journal, because the serial record contains subject representations of this journal, Field 050 shows a Library of Congress Classification number ("TK6563") that

Google Exhibit No. 1063
Page 0133

represents general work on “Radio receiving apparatus” (source: Library of Congress Classification Schedule T at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), Field 082 shows a Dewey Decimal Classification number (“621.38”) that represents “Electronics, communications engineering” (source: Dewey Decimal Classification Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf), and Fields 650 show four Library of Congress subject headings, with “Household electronics,” “Radio |x Receivers and reception” (“|x” indicates a topical subheading), “Television |x Receivers and reception,” and “Electronic apparatus and appliances” as the main topics, each followed by “Periodicals” as a form subheading, encoded in subfield “v”, to indicate the main topics are treated in a journal.

290. This MARC record (**Ex. 1040B**) shows that *IEEE Transactions on Consumer Electronics* is a long-running journal that began publication in 1975. The MARC record makes this journal discoverable, and users can discover it by the journal’s ISSN, CODEN, title, variant title, three organizations and a conference name. They can also conduct subject searches to discover this journal.

C. Summary of My Opinion on *Lee*

291. The date stamp on **Ex. 1040A** shows the Library of Congress received the issue containing *Lee* on "OCT 31 2002." As discussed above, a newly received journal issue usually becomes publicly accessible within a week after serial check-in

Google Exhibit No. 1063
Page 0134

is completed. It is therefore my opinion that the issue containing *Lee* was publicly accessible at the Library of Congress by November 7, 2000. The MARC record for the journal containing *Lee* was first created on October 21, 1975 (**Ex. 1040B**), and it shows interested users could discover this journal at the Library of Congress by conducting subject searches and by searching by the journal's ISSN, CODEN, title, and variant title.

XXIII. Public Availability of *De Smet 2004* (Ex. 1042)

292. **Exhibit 1042** is a true and correct copy of "Do not zero-pute: an efficient homespun MPEG-audio layer II decoding and optimization strategy," ("*De Smet 2004*"), by De Smet et al. in *2004 Multimedia Conference MM'04*, pp. 376-379, that I received from counsel. This copy of *De Smet 2004* is presented as **Ex. 1042** in this declaration.

293. Page 1 (internal page 376) shows the title, authors, abstract, categories and subject descriptors, and general terms. The bottom of the page shows this article is in "MM'04, October 10-16, 2004, New York, New York, USA" and has a "2004" copyright date with ACM as the copyright holder. It also shows this article is in a work whose International Standard Book Number (ISBN) is "1-58113-893-8." **Ex. 1042** shows *De Smet 2004* has four pages in eight sections, including one figure, two tables, four equations, and seven references.

Google Exhibit No. 1063
Page 0135

A. Records of British Library

294. To determine the public availability date of *2004 Multimedia Conference MM'04*, I searched WorldCat.org, the free web portal, and a record at https://search.worldcat.org/title/1063590195 showed the British Library held this work. This WorldCat record and a record of the British Library that I obtained from OCLC FirstSearch are presented as **Ex. 1042A** in this declaration. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

295. Page 1 of **Ex. 1042A** is the WorldCat record that shows the title, subtitle ("MM'04"), format, language, publisher ("ACM Press"), publication year ("2004"), ISBNs, OCLC number ("1063590195"), and subjects, and indicates the British Library holds this work.

296. Pages 2 and 3 are the bibliographic record for *2004 Multimedia Conference: MM'04* that I obtained from the OCLC FirstSearch, which is a search system that enables users to explore propriety databases, including OCLC's WorldCat database. As discussed earlier, the general public can use the free web portal, WorldCat.org, to explore the collections of more than 10,000 libraries worldwide. In addition, subscribers to OCLC FirstSearch can search the WorldCat database. The search results from these two systems are largely the same, and the main difference is records from OCLC FirstSearch contain more technical details,

Google Exhibit No. 1063
Page 0136

such as "Date of Entry." According to *OCLC Glossary*, "Date of Entry" in FirstSearch detailed records (https://help.oclc.org/Librarian_Toolbox/OCLC_glossaries/OCLC_glossary#D) "is the label for MARC field 008/00-05" (meaning the first six digits of Field 008 that indicate the date when a record was entered into the WorldCat database).

297. Pages 2 and 3 show the title and subtitle *2004 Multimedia Conference* and indicates the British Library as the holding library. The record details show the title, corporate author, publisher, publication year, physical description, language, ISBN, and subjects ("Multimedia," "MM," and "ACM"). The "Entry" field and the "Database" field show *2004 Multimedia Conference* was entered into the WorldCat database on "20050315" (*i.e.*, March 15, 2005). The "Accession No" field shows the record's OCLC number is "1063590195," which is the same as the record number for this work that I found in WorldCat.org.

B. Summary of My Opinion on *De Smet 2004*

298. Based on the WorldCat.org record, and the OCLC FirstSearch record (**Ex. 1042A**), it is my opinion that the British Library cataloged *2004 Multimedia Conference* that contains *De Smet 2004* on March 15, 2005, and entered the record into the OCLC WorldCat database. As of that date, the British Library record became available to OCLC members for cataloging and collection development purposes. In addition, the record became discoverable for subscribers to the OCLC FirstSearch

Google Exhibit No. 1063
Page 0137

system, and for the general public who can search the web portal, WorldCat.org, for free. interested users could discover the record by subject searches of the subject terms. They could also discover this work by the title, corporate author, and ISBNs of this work.

XXIV. Public Availability of *De Smet* (Ex. 1043)

299. **Ex. 1043** is a true and correct copy of "Optimized recursive subband synthesis windowing for implementing efficient MPEG audio decoders," ("*De Smet*"), by De Smet and Bruyland, *IEEE Signal Processing Letters*, vol. 10, no. 10, 303–306, that I received from counsel. This journal issue has an issue date of "October 2003." This copy is presented as **Ex. 1043** in this declaration.

300. Page 1 (internal page 303) shows the title, authors, abstract, and index terms. The bottom of the page shows *De Smet* appears in a journal whose International Standard Serial Number (ISSN) is "1070-9908" and this article has a "2003" copyright date and IEEE (Institute of Electrical and Electronics Engineers, Inc.) holds the copyright. **Ex. 1043** shows *De Smet* has a total of four pages in nine sections that includes six equations, and 15 references.

A. Date Stamped Copy from Library of Congress

301. To determine the public availability of *De Smet*, I searched WorldCat.org, the free web portal, to identify libraries holding the journal containing *De Smet*, and learned that the Library of Congress was one of the holding libraries.

Google Exhibit No. 1063
Page 0138

After verifying the holdings information in their online catalog, I requested a copy of *De Smet* from the Library of Congress through a document vendor. This library copy of *De Smet* is presented as **Ex. 1043A** in this declaration.

302. Page 1 of **Ex. 1043A** shows the spine of a bound volume identified as "10, 2003" of "IEEE Signal Processing Letters" that has a call number of "TK 5102.9 .I32." Page 2 is the issue cover that shows this journal is a publication of the IEEE Signal Processing Society, its ISSN is 1070-9908 and its CODEN (a unique six-character code that identifies journals) is "ISPLEM." The cover also identifies this issue as volume 10, number 10 that has a "October 2003" issue date. The table of contents on this cover page shows *De Smet* begins on page 303. Page 3 is the copyright page that shows this journal is published monthly by the Institute of Electrical and Electronics Engineers, Inc., and this issue has a "2003" copyright date with the publisher as the copyright holder. Page 4 is a page that shows a date stamp of "Library of Congress NOV 10 2003 Copyright Office" that indicates this issue was received by the Library of Congress on that date. Pages 5 to 8 (internal pages 303 to 306) are the full text of *De Smet*.

303. I have compared **Ex. 1043** with the Library of Congress copy of *De Smet* (**Ex. 1043A**) and found them to have the same content. The date stamp on **Ex. 1043A** indicates Library of Congress received this issue containing *De Smet* on November 10, 2003. As discussed above, a customary library practice is to document

Google Exhibit No. 1063
Page 0139

the receipt of a new journal issue by placing the issue through the serial check-in system. After that, the new issue is promptly processed for public access, usually within a week after serial check-in is completed. It is therefore my opinion that the issue containing *De Smet* became publicly accessible at the Library of Congress by November 17, 2003.

B. Library of Congress Records

304. As evidence that interested users could discover the journal containing *De Smet* at the Library of Congress to access this article, I obtained serial records for *IEEE Signal Processing Letters* from the online catalog of the Library of Congress. **Ex. 1043B** is a true and correct copy of the bibliographic and MARC records for this journal. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

305. Pages 1 to 3 of **Ex. 1043B** are the bibliographic record and Pages 4 to 6 are the MARC record for this journal. Field 008 of the MARC record shows the serial record was created on "930727" (*i.e.*, July 27, 1993), and the code of "c19949999" following the record creation date indicates this publication is a continuing publication that began in 1994 and has no end date for publication yet. Subfields "a" and "c" of Field 040 show "NSD" (the National Serial Data Program at the Library of Congress) created the original record and subfield "d" shows "DLC"

Google Exhibit No. 1063
Page 0140

(Library of Congress) modified the record later. Field 022 shows the ISSN, Field 030 shows the CODEN, Field 245 shows the journal title, and Field 260 shows IEEE began publishing this journal in 1994. These details match the information presented in **Ex. 1043A**, confirming this serial record represents the journal containing *De Smet*.

306. The serial record includes several ways for users to discover this journal, including the ISSN, CODEN, journal title, and two variant titles in Field 246. In addition, the parent organization of this journal in included in Field 710 to help users discover this journal.

307. Interested users could also conduct subject searches to discover this journal, because the serial record contains subject representations of this journal, Field 050 shows a Library of Congress Classification number ("TK5102.9") that represents "Signal processing" (source: Library of Congress Classification Schedule T. at https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf), Field 082 shows a Dewey Decimal Classification number ("621.382/2/05") that represents journals on "Communications engineering" (source: Dewey Decimal Classification Class 600 at https://www.oclc.org/content/dam/oclc/webdewey/help/600.pdf), and Fields 650 show three Library of Congress subject headings, with "Signal processing," "Image processing," and "Speech processing systems" as the main topics, each followed by "Periodicals" as a form subheading, encoded in subfield "v", to indicate the main topics are treated in a journal.

308. This MARC record (**Ex. 1043B**) shows that *IEEE Signal Processing Letters* is a long-running journal that began publication in 1994. The MARC record makes this journal discoverable, and users can discover it by the journal's ISSN, CODEN, title, variant titles, and the parent organization of the journal. They can also conduct subject searches to discover this journal.

C. Copyright Registration Record

309. **Ex. 1043C** is a true and correct copy of the copyright registration record for *IEEE Signal Processing Letters*. I personally located, identified, and obtained this record from the public catalog of the US Copyright Office. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

310. The title, serial publication year, imprint, ISSN, frequency, and copyright claimant match the information presented in **Ex. 1043A**, confirming this record represents *IEEE Signal Processing Letters*. The "Issues Registered" field shows v. 10, no. 10, which contains *De Smet*, has a publication date of "2003-09-18" (*i.e.*, September 18, 2003). This date aligns well with the date stamp on **Ex. 1043A** that shows the Library of Congress received v. 10, no. 10 on November 10, 2003.

D. Summary of My Opinion on *De Smet*

311. The copyright registration record shows v. 10, no. 10 of *IEEE Signal Processing Letters* that contains De Smet was published on September 18, 2003 (**Ex. 1043C**). The date stamp on **Ex. 1043A** shows the Library of Congress received this

Google Exhibit No. 1063
Page 0142

issue on "NOV 10 2003." As discussed above, a newly received journal issue usually becomes publicly accessible within a week after serial check-in is completed. It is therefore my opinion that the issue containing *De Smet* was publicly accessible at the Library of Congress by November 17, 2003. The MARC record for the journal containing *De Smet* was first created on July 27, 1993 (**Ex. 1043B**), and it shows interested users can discover this journal at the Library of Congress by conducting subject searches and by searching by the journal's ISSN, CODEN, title, and variant titles.

XXV. Public Availability of *MPEG Overview* (Ex. 1044)

312. **Ex. 1044** is a true and correct copy of MPEG-4 Overview (V. 21) edited by Rob Koenen, identified as N4668, and published in "March 2002" that I received from counsel. This copy is presented as **Ex. 1044** in this declaration.

313. Page 1 is the cover sheet that shows the source ("WG11 (MPEG)"), status ("Final"), title ("MPEG-4 Overview – (V.21 – Jeju Version)"), and editor ("Rob Koenen"), and identifies this document as "ISO/IEC JTCI/SC29/WG11 N4668" that was published in "March 2022." The top of the page shows "International Organisation for Standardisation" and "ISO/IEC JTCI/SC29/WG11 Coding of Moving Pictures and audio." The footer shows "MPEG-4 Overview," "1999-2002" as the copyright dates, and a note on access ("unlimited reproduction permitted if not modified").

Google Exhibit No. 1063
Page 0143

314. Page 2 is the Executive Overview that explains the history and development of MPEG-4, identifies resources for further information, and indicates this document "gives an overview of the MPEG-4 standard, explaining which pieces of technology it includes and what sort of applications are supported by this technology." Pages 3 and 4 are the table of contents that shows *MPEG Overview* has 13 sections and three annexes. **Ex. 1044** shows *MPEG Overview* has a total of 77 pages, including 24 figures (some color), one table in Annex A, and a glossary in Annex C.

A. Internet Archive Records

315. Internet Archive is a digital library that archives scholarly materials, cultural artifacts, and other web resources on the Internet and provides the public with free access to its collections. I have used the Internet Archive in the past and am familiar with their practice of assigning an extended URL to an archived resource in the format of http://www.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] to show the point of time the archived file was captured. Since Internet Archive usually does not archive web resources the moment they appear on the Internet, and a web resource must be available first before it can be archived, the archive date of a file captured by the Internet Archive is typically later than the date when the file first became publicly accessible. In other words, the archive date means the archived file was available before, and definitely

Google Exhibit No. 1063
Page 0144

no later than, the archive date.

316. To determine the public availability of *MPEG Overview*, I searched Internet Archive for archived information on this document and found that *MPEG Overview* was publicly accessible by August 1, 2003. My discovery is discussed below.

317. **Ex. 1044A** is a true and correct copy of MPEG-4 Overview (v.21) edited by Rob Koenen, which is identified as ISO/IEC JTC1/SC29/WG11 N4668 published in March 2002 that I personally located, identified, and obtained from Internet Archive. This is the type of material experts in my field would reasonably rely upon when forming their opinions.

318. Page 1 is a screen capture of a Wayback Machine calendar that shows this copy was archived 586 times from "1 August 2003" to "4 Dec 2024" and this particular copy was archived on "AUG 01 2003" and available at https://web.archive.org/web/20030801081401/http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm. I have closely compared **Ex. 1044A** and **Ex. 1044** and found them to contain the same 24 figures and the same table (in Annex A). The table of contents in these two copies are slightly different in that the archived copy shows subheadings for Section 4 and Section 13, and the heading for Section 13 is different from that in **Ex. 1044**. Other than these differences and formatting difference, I have found **Ex. 1044** and the archived copy (**Ex. 1044A**) to have the same content.

Google Exhibit No. 1063
Page 0145

B. Summary of My Opinion on *MPEG Overview*

319. Based on the documents I have reviewed, it is my opinion that MPEG-4 Overview (v.21) edited by Rob Koenen, which is identified as ISO/IEC JTC1/SC29/WG11 N4668 published in March 2002 (**Ex. 1044**), was publicly accessible by August 1, 2003, on the Internet.

XXVI. Public Availability of *Mesarovic* (Ex. 1047)

320. **Exhibit 1047** is a true and correct copy of "Optimized Huffman Decoding for AAC," ("*Mesarovic*"), by Mesarovic et al. in *Audio Engineering Society Conference: UK 16th Conference: Silicon for Audio*, pp. 117-122, published by the Audio Engineering Society, that I received from counsel. This copy of *Mesarovic* is presented as **Ex. 1047** in this declaration.

321. Page 1 (internal page 117) shows the title, authors and abstract. The bottom of the page shows this article is included in "Silicon for Audio – AES 16TH UK Conference 2001." **Ex. 1047** shows *Mesarovic* has six pages in three sections, including three tables and five references.

A. Records of German National Library of Science and Technology

322. To determine the public availability date of *Silicon for Audio* that contains *Mesarovic*, I searched WorldCat.org, the free web portal, and a record at https://search.worldcat.org/title/249633743 showed Technische Informationsbibliothek (TIB), the German National Library of Science and

Google Exhibit No. 1063
Page 0146

Technology, held this work. I then obtained TIB's records and table of contents for this work from the TIB online catalog, and also obtained the TIB record in OCLC FirstSearch to show the date of public availability. These records are presented in **Ex. 1047A** in this declaration. I personally located, identified, and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions.

323. Page 1 of **Ex. 1047A** is the WorldCat record that shows the title, subtitle, authors, physical description, OCLC number, and "Inhaltsverzeichnis" (table of contents in German) of this work. The record also shows TIB is the holding library. Page 2 is the bibliographic record for *Silicon for Audio* obtained from the online catalog of TIB. The record shows the title, subtitles, personal author, corporate author, conference information ("UK conference. Audio Engineering Society (AES); 16 (London): 2001.04.09-10"), language, publisher information and publication year, pagination, link to the table of contents (http://www.gbv.de/dms/tib-ub-hannover/358253993.pdf), call number of RS 2572(16), status as "nicht verleihbar" ("not available for loan" per Google Translate), availability as "derzeit verfügbar" ("currently available" per Google Translate), and interested users could request access to this work. Following the link for the table of contents, I obtained the cover and table of contents of *Silicon for Audio* and presented them as pages 3 and 4 of **Ex. 1047A**. The cover shows the conference name, conference chairman, dates, and

Google Exhibit No. 1063
Page 0147

location, and the table of contents shows *Mesarovic* begins on page 117 of this work.

324. Since TIB does not present their record in MARC in their online catalog, I have located their record from OCLC FirstSearch, which is a search system that enables users to explore propriety databases, including OCLC's WorldCat database. As discussed earlier, the general public can use the free web portal, WorldCat.org, to explore the collections of more than 10,000 libraries worldwide. In addition, subscribers to OCLC FirstSearch can search the WorldCat database. The search results from these two systems are largely the same, and the main difference is records from OCLC FirstSearch contain more technical details, such as "Date of Entry." According to *OCLC Glossary*, "Date of Entry" in FirstSearch detailed records (https://help.oclc.org/Librarian_Toolbox/OCLC_glossaries/OCLC_glossary#D) "is the label for MARC field 008/00-05" (meaning the first six digits of Field 008 that indicate the date when a record was entered into the WorldCat database).

325. Pages 5 and 6 are the FirstSearch record for *Silicon for Audio: Audio Engineering Society 16th UK Conference*, and "UNIV Hannover & TIB" is listed as the holding library. The record details show the title, author, corporate author, conference author, publisher, publication year, pagination, language, and a link to the table of contents. The "Date of Entry" field and the "Database" field show *Silicon for Audio* was entered into the WorldCat database on "20021218" (*i.e.*, December 18, 2002). The "Accession No" field shows the record's OCLC number is "249633743,"

which is the same as the number for this work that I found in WorldCat.org.

B. Summary of My Opinion on *Mesarovic*

326. Based on the WorldCat.org record, the TIB bibliographic record and the table of contents provided through this record, and the OCLC FirstSearch record (**Ex. 1047A**), it is my opinion that TIB cataloged *Silicon for Audio* that contains *Mesarovic* on December 18, 2002, and entered the record into the OCLC WorldCat database. As of that date, the TIB record became available to OCLC members for cataloging and collection development purposes. In addition, the record became discoverable for subscribers to the OCLC FirstSearch system, and for the general public who can search the web portal, WorldCat.org, for free. Interested users could discover this work by the title, conference chair, conference author, and conference information.

Google Exhibit No. 1063
Page 0149

XXVII. Conclusion

327. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in this proceeding. If required, I will appear for cross-examination at the appropriate time.

328. I hereby declare that all statements made herein of my own knowledge are true and that all statements are made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. §1001.

Dated: 3-13-2025

Respectfully submitted,

Ingrid Hsieh-Yee

Ingrid Hsieh-Yee, Ph.D.
Annapolis, MD

Google Exhibit No. 1063
Page 0150

Appendix A

Ingrid Hsieh-Yee

Professor Emerita
Dept. of Library and Information Science
Catholic University of America
Washington, D.C. 20064
E-mail: hsiehyee@cua.edu

Education

Ph.D. 1990. Library and Information Studies, University of Wisconsin-Madison
Minors: Sociology and Psychology

M.A. 1984. Library and Information Studies, University of Wisconsin-Madison.

M.A. 1983. Comparative Literature, University of Wisconsin-Madison.

B.A. 1980. Foreign Languages and Literature, National Taiwan University.

Work Experience

Professor, School/Dept. of Library and Information Science, Catholic University of America, 1990-2022 (Assistant Professor, 1990-1996; Associate Professor, 1997-2004; Professor 2004-2022; Professor Emerita 2022)

Co-Chair, Dept. of Library and Information Science, Catholic University of America, June 2015-August 2016.

Acting Dean, School of Library and Information Science, Catholic University of America, January 2010-June 2012.

Cataloger, Dept. of Legislative Reference Library, Annapolis, Maryland, 1989-1990.

Lecturer, School of Library and Information Studies, University of Wisconsin-Madison, 1988.

Teaching Assistant, School of Library and Information Studies, University of Wisconsin-Madison, 1986-1988.

Cataloger, Health Sciences Library, University of Wisconsin-Madison, 1984-1986.

Areas of Teaching and Research Interests

Information Organization and Access; Metadata; Cataloging & Classification; Information Architecture; Information Retrieval; Digital Collections; Social Media; Scholarly

Communication; Information Behavior; Health Informatics; Human Computer Interaction; Usability Studies

Grants & Honors

Cultural Heritage Information Management Project. IMLS grant. Amount: $498,741. Period: Aug. 2012 to July 2015. Co-PI with Dr. Youngok Choi.

D.C. Health Information Technology (HIT4): Building Capacity & Providing Access in Our Nation's Capital. Dept. of Labor H2B Training Grant. Grant amount: $4,175,500. Grant period: Nov. 2011 to Dec. 2015. Partner with the Metropolitan School of Professional Studies of the Catholic University of America, Children's National Medical Center, D.C. Department of Employment Services, Holy Cross Hospital, Howard University, Center for Urban Progress, Providence Hospital, and Sibley Memorial Hospital.

Capital Health Careers Project. Department of Labor Healthcare Sector and Other High Growth and Emerging Industries Grant. Grant amount: $4,953,999. Grant period: March 2010 – February 2013. Awarded to a group of healthcare organizations and educational institutions in Washington, D.C. Providence Health Foundation of Providence Hospital (Lead institution). Part of the grant supported the development of a Master's degree program in Information Technology with a concentration in Health Information Technology offered by the School of Library and Information Science.

The Washington D.C. School Librarians Project. IMLS grant. Grant amount: $412,660. Grant period: Aug. 2007 – June 2011. The School partnered with the District of Columbia Public Schools (DCPS) and the District of Columbia Library Association to educate and mentor school media specialists for the DCPS system. PI, Jan. 2010 to June 2011.

SIG Member of the Year, American Society for Information Science and Technology (2009).

Most Outstanding Paper of *OCLC Systems & Services* (2001).

ALISE Research Grant (2001).

Most Outstanding Paper of *OCLC Systems & Services* (2000).

Research Grant from ERIC (1999-2000).

Best Research Paper Award; Association for Library and Information Science Education (1998).

Research Grants, Catholic University of America. 1991, 1992, 1993, 1996, 1998, 1999, 2004, 2005, 2006, 2007, 2013-14.

Cooperative Faculty Research Grant, Consortium of Universities in the Washington Metropolitan Area (1993-1994).

Cooperative Research Grant, Council on Library Resources (1993-1994).

Journal of the American Society for Information Science Best Paper Award (1993).

ASIS/ISI Information Science Doctoral Dissertation Scholarship (1989).

HEA Title IIB Fellowship (Dept. of Education) (1989)

Chinese-American Librarians Association Scholarship (1987).

Beta Phi Mu (1985).

Vilas Fellowship, University of Wisconsin-Madison. 1984

Publications

Hsieh-Yee, I. (2021). Can We Trust Social Media? *Internet Reference Services Quarterly*, 1-15. Published online: 28 July, 2021. 10.1080/10875301.2021.1947433.

Bailey, T., & Hsieh-Yee, I. (2020). Combating the Sharing of False Information: History, Framework & Literacy Strategies, *Internet Reference Services Quarterly*, 24 (1-2), 9-30.

Choi, Y., and Hsieh-Yee, I. (2010). Finding Images in an OPAC: Analysis of User Queries, Subject Headings, and Description Notes. *Canadian Journal of Information and Library Science,* 34(3): 271 – 295.

Hsieh-Yee, I. (2008). Educating Cataloging Professionals in a Changing Information Environment. *Journal of Education for Library and Information Science,* 46(2): 93-106.

Vellucci, S. L., Hsieh-Yee, I., and Moen, W.E. (2007). The Metadata Education and Research Information Commons (MERIC): A Collaborative Teaching and Research Initiative. *Education for Information,* 25(3&4): 169-178.

NISO Framework for Guidelines for Building Good Digital Collections. 3rd ed. Baltimore, MD: National Information Standards Organization, 2007. Also available online: http://www.niso.org/framework/framework3.pdf (NISO Working Group members: Priscilla Caplan (chair), Grace Agnew, Murtha Baca, Tony Gill, Carl Fleischhauer, Ingrid Hsieh-Yee, Jill Koelling, and Christie Stephenson.)

Choi, Y., Hsieh-Yee, I., and Kules, B. (2007). Retrieval Effectiveness of TOC and LCSH. *Proceedings of the Joint Conference on Digital Libraries*, pp. 233-234.

Vellucci, S., and Hsieh-Yee, I. (2007). They Didn't Teach Me That in Library School! Building

a Digital Teaching Commons to Enhance Metadata Teaching, Learning and Research. *Proceedings of the National Conference of the Association of College and Research Libraries, Baltimore, MD*, pp. 26-31.

Mitchell, Vanessa, and Ingrid Hsieh-Yee. (2007). Converting Ulrich's Subject Headings to FAST Headings: A Feasibility Study. *Cataloging & Classification Quarterly,* 45(1): 59-85.

Hsieh-Yee, I., Tang, R., and Zhang, S. (2007). User Perceptions of a Federated Search System. *IEEE Technical Committee on Digital Libraries Bulletin,* Summer 3(2) (URL = http://www.ieee-tcdl.org).

Tang, R., Hsieh-Yee, I., and Zhang, S. (2007). User Perceptions of MetaLib Combined Search: An Investigation of How Users Make Sense of Federated Searching." *Internet Reference Services Quarterly,* 12(12): 211-236.

Hsieh-Yee, I., Tang, R., and Zhang, S. (2006). User Perceptions of a Federated Search System. *Proceedings of the Joint Conference on Digital Libraries*, *June 11-15, 2006, Chapel Hill*, p. 338.

Hsieh-Yee, I. (2006). *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide*. 2nd ed. Westport, Conn.: Libraries Unlimited.

NISO A Framework of Guidance for Building Good Digital Collections. 2nd ed. Bethesda, MD: National Information Standards Organization, 2004. Framework Advisory Group: Grace Agnew, Liz Bishoff, Priscilla Caplan (Chair), Rebecca Gunther and Ingrid Hsieh-Yee.

Hsieh-Yee, I. (2004). Cataloging and Metadata Education in North American LIS Programs. *Library Resources & Technical Services,* 48(1): 59-68.

Hsieh-Yee, I. (2004). Cataloging and Metadata Education. In Gary E. Gorman (Ed.), *International Yearbook of Library and Information Management 2003: Metadata Applications and Management*, (pp.204-234). London: Facet Publishing.

Yee, P. L., Hsieh-Yee, I., Pierce, G.R., Grome, R., and Schantz, L. (2004). Self-Evaluative Intrusive Thoughts Impede Successful Searching on the Internet. *Computers in Human Behavior,* 20(1): 85-101.

Hsieh-Yee, I. (2003). Cataloging and Metadata Education: A Proposal for Preparing Cataloging Professionals of the 21st Century. A report submitted to the ALCTS-Education Task Force in response to Action Item 5.1 of the *Bibliographic control of Web Resources: A Library of Congress Action Plan*. Approved by the Association for Library Collections and Technical Services. Web version available since April 2003 at http://lcweb.loc.gov/catdir/bibcontrol/CatalogingandMetadataEducation.pdf.

Hsieh-Yee, I. (2002). Cataloging and Metadata Education: Asserting a Central Role in Information Organization. *Cataloging & Classification Quarterly* 34(½): 203-222.

Hsieh-Yee, I., and Smith, M. (2001). The CORC Experience: Survey of Founding Libraries, Part I. *OCLC Systems & Services,* 17: 133-140. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2001" award.)

Hsieh-Yee, I., and Smith, M. (2001). The CORC Experience: Survey of Founding Libraries, Part II, Automated Tools and Usage. *OCLC Systems & Services,* 17: 166-177. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2001" award.)

Hsieh-Yee, I. (2001). ERIC User Services: Changes and Evaluation for the Future. *Government Information Quarterly,* 18: 31-42.

Hsieh-Yee, Ingrid. (2001). Research on Web Search Behavior. *Library and Information Science Research,* 23: 167-185.

Logan, E., and Hsieh-Yee, I. (2001). Library and Information Science Education in the Nineties. *Annual Review of Information Science and Technology,* 35: 425-477.

Hsieh-Yee, I. (Ed.) (2001). *Library and Information Science Research*, 23 (2). A special issue in honor of the retirement of Douglas L. Zweizig.

Hsieh-Yee, I. (2000). *ERIC User Services: Evaluation in a Decentralized Environment*. Washington, D.C.: Dept. of Education.

Hsieh-Yee, Ingrid. (2000). *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide*. Littleton, CO: Libraries Unlimited.

Hsieh-Yee, I. (2000). Organizing Internet Resources: Teaching Cataloging Standards and Beyond. *OCLC Systems & Services,* 16: 130-143. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2000" award.)

Hsieh-Yee, I. (1998). The Retrieval Power of Selected Search Engines: How Well Do They Address General Reference Questions and Subject Questions? *Reference Librarian,* 60: 27-47.

Hsieh-Yee, I. (1998). Search Tactics of Web Users in Searching for Texts, Graphics, Known Items and Subjects: A Search Simulation Study. *Reference Librarian,* 60: 61-85. (Received the 1997 Best ALISE Research Paper Award.)

Hsieh-Yee, I. (1997). Access to OCLC and Internet Resources: LIS Educators' Views and Teaching Practices. *RQ,* 36: 569-86.

Hsieh-Yee, I. (1997). Teaching Online and CD-ROM Resources: LIS Educators' Views and Practices. *Journal of Education for Library and Information Science,* 38: 14-34.

Hsieh-Yee, I. (1996). The Cataloging Practices of Special Libraries and Their Relationship with OCLC. *Special Libraries,* 87: 10-20.

Hsieh-Yee, I. (1996). Modifying Cataloging Practice and OCLC Infrastructure for Effective Organization of Internet Resources. In *Proceedings of the OCLC Internet Cataloging Colloquium*. [Online]. Available: http://www.oclc.org/oclc/man/colloq/hsieh.htm

Hsieh-Yee, I. (1996). Student Use of Online Catalogs and Other Information Channels. *College & Research Libraries,* 57: 161-175.

Hsieh-Yee, I. (1995). Ten entries in James S. C. Hu (Ed.), *Encyclopedia of Library & Information Science*, 913, 1028-29, 1036, 1037, 1145-46, 1514, 1575, 1763-64, 2216-27, 2378-79. Taipei, Taiwan: Sino-American Publishing. (Topics include "Advanced Technology/Libraries," "Information Ethics," "Instruction on Cataloging and Classification," "Instruction on Reference Services.")

Hsieh-Yee, I. (1993). Effects of Search Experience and Subject Knowledge on Online Search Behavior: Measuring the Search Tactics of Novice and Experienced Searchers. *Journal of the American Society for Information Science,* 44: 161-174. (Received the 1993 Best JASIS Paper Award.)

Works in Progress

Hsieh-Yee, I. Actionable lessons for debunking false information: A comparative analysis of literacy lessons in communication and library and information science

Hsieh-Yee, I. Representing diverse voices and communities in cataloging and indexing: The evolution of controlled vocabularies

Hsieh-Yee, I. Information Organization and Retrieval in Practice (a monograph scheduled for completion and publication in 2023)

Presentations

Bailey, T. & Hsieh-Yee, I. (February 2020). The Phenomena of Sharing Misinformation and the Need for Information Literacy. Presented at the 2020 Bridging the Spectrum Symposium, Washington, D.C., February 14, 2020.

Hsieh-Yee, I. and Fragan-Fly, J. (May 2018) Trends, Design & Strategies for Digital Scholarship Services. Presented at the 2018 Maryland/Delaware Library Association Conference, Cambridge, MD.

Hsieh-Yee, I. (February, 2018) Research Data Management: What It Takes to Succeed. Presented at the 10th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. (February, 2017) *Research Data Management: New Competencies and Opportunities for Information Professionals*. Presented at the 9th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. and Lawton, P. (February, 2017) *Enhancing Catholic Portal Searches with User Terms and LCSH*. Presented at the 9th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. (2016, October) *Visualizing Data for Information*. Presented at the 2016 Virginia Library Association Conference, Hot Springs, VA.

Hsieh-Yee, I. (2016, August) *Religious Materials Toolbox for Archivists: Solutions to Problems Facing the Profession*. Presented at Archives * Records 2016, Atlanta, GA.

Hsieh-Yee, I. and Lawton, P. (2016, March) *Enhancing Retrieval of Catholic Materials with LCSH Knowledge Structure*. Presented at the 2016 Catholic Library Association Conference, San Diego, CA.

Fagan-Fry, J. and Hsieh-Yee, I. (2016, February) *Approaches to Digital Scholarship at Top Universities around the World: Scholarly Publishing in the Digital Age.* Presented at the 8th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee and Fagan-Fry, J. (2016, January) *Innovative Services for Digital Scholarship at Top 100 Research Libraries of the World*. Poster presented at the 2016 Annual Conference of the Association for Library and Information Science Education, Boston, Mass.

Hsieh-Yee, I. and Lawton, P. (2015, June). *Crowdsourcing terms for CRRA portal themes.* Poster presented at the third CRRA symposium and annual meeting, Bringing the created toward the Creator: Liturgical art and design since Vatican II. Catholic Theological Union, Chicago, Illinois.

Hsieh-Yee, I. and Lawton, P. (2015, February). *Crowdsourcing terms for thematic exploration in the Catholic Portal.* Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I., James, R., and Fagan-Fry, J. (2015, February). *Support for digital scholarship at top university libraries of the world*. Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I., Zhang, S., Lin, K., and Cherry, S. (2015, February). *Thus said the end users: Summon experience and support for research workflows*. Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.

Yontz, E., Hsieh-Yee, I., & Houston, S. (2015, February). *Healthy Heroes Summer Reading Club: Developing healthy youth at public libraries.* 11th Annual Jean Mills Health Symposium, Greenville, North Carolina.

Yontz, E., Hsieh-Yee, I., and Houston, S. (2015, January). *Healthy youth and libraries: A pilot study.* Association for Library & Information Science Education (ALISE) Annual Conference, Chicago, Illinois.

Hsieh-Yee, I. (2014, May). *Linking CRRA resources to portal themes via authority files*. Presented at the Catholic Research Resources Alliance 2014 Membership Meeting, Marquette, WI.

Hsieh-Yee, I. (2014, April). *Enhancing subject access to CRRA resources*. Presented at the 2014 Catholic Library Association Conference, Pittsburgh, PA.

Hsieh-Yee, I. (2014, January). *Health Information Technology Program: Educational entrepreneurship in action*. Presented at the 2014 annual Conference of the Association for Library and Information Science Education, Philadelphia, PA

Hsieh-Yee, I., Zhang, S., Lin, K., and Cherry, S. (2014, January). *Discovering information through Summon: An analysis of user search strategies and search success*. Paper presented at the 6th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. (2012, December). *National Digital Stewardship Alliance and SLIS at CUA: An Educational Partnership*. Paper presented at Best Practices Exchange: Acquiring, Preserving, and Providing Access to Government Information in the Digital Era, Annapolis, MD

Choi, Y. and Hsieh-Yee, I. (2010, January). *Finding Images in an OPAC: Analysis of User Queries, Subject Headings, and Description Notes*. Paper presented at 2nd Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. and Coogan, J. (2010, January). *Google Scholar vs. Academic Search Premier: What Libraries and Searchers Need to Know*. Paper presented at 2nd Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (2009, November). *Information Science Education: An LIS School's Perspective*. Paper presented at Annual Meeting of the American Society for Information Science and Technology, Vancouver, British Columbia, Canada.

Hsieh-Yee, I., Menard, E., Ya-Ning Chen, A., Shu-Jiun Chen, S., Kalfatovic, M. R., Wisser. K. M. (2009, November). *Information Organization in Libraries, Archives and Museums: Converging Practices and Collaboration Opportunities*. Presented at Annual Meeting of the American Society for Information Science and Technology, Vancouver, British Columbia, Canada. (Organizer and moderator of this panel.)

Hsieh-Yee, I. and Coogan, J. (2009, July). *Catching up to Google Scholar: The Retrieval Power of Academic Search Premier and Google Scholar*. Poster presented at American Library Association Conference, Chicago, Illinois.

Hsieh-Yee, I., with the CUA Scholarly Communications Project Team. (2009, January). *Digital Scholarship@CUA: Developing an Institutional Repository for CUA*. Poster presented at 1st Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Wise, M., Cylke, K., and Hsieh-Yee, I. **(**2009, January). *Digital Talking Books: Meeting the Needs of the Blind and the Handicapped*. Paper presented at the Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (2009, January). *User Expectations of MERIC*. Presented at the Information Organization Competencies for the 21st Century Discussion Session of the 2009 Conference of the Association for Library and Information Science Education, Denver, Colorado.

Choi, Y., and Hsieh-Yee, I. (2008,November). *Subject Access for Images in an OPAC.* Annual Meeting of the American Society for Information Science and Technology, Columbus, Ohio. (Also co-organized a panel on Retrieving and Using Visual Resources: Challenges and Opportunities for Research and Education.)

Hsieh-Yee, I. (2008, June). *Educating Cataloging Professionals in a Changing Information Environment*. National Taiwan University, Taipei, Taiwan.

Vellucci, S. L., Moen, W.E., Hsieh-Yee, I., Marson, B., and Wisser, K. (2008, January) *Building a Metadata Education and Research Community through MERIC (Metadata Education and Research Information Commons): Demo and Stakeholder Input*. A panel presented at the 2008 Conference of the Association for Library and Information Science Education, Philadelphia, Pennsylvania.

Hsieh-Yee, I., Choi,Y. and Kules, B. (2007, October). *Searching for Books and Images in OPAC: Effects of LCSH, TOC and Subject Domains.* A poster presented at the American Society for Information Science and Technology Annual Meeting, Milwaukee, Wisconsin.

Hsieh-Yee, I. and Coogan, J. (2007, August) *Google Scholar vs. Academic Search Premier: A Comparative Analysis.* Presented to the Faculty and Staff of the University of the District

of Columbia.

Hsieh-Yee, I. and Coogan, J. (2007, June). *Google Scholar vs. Academic Search Premier: A Comparative Analysis*. Presented to the Washington Research Library Consortium Community, Catholic University of America, Washington, D.C.

Hsieh-Yee, I., Choi, Y., and Kules, B.. (2007, June). *What Users Need for Subject Access: Table of Contents or Subject Headings?* A poster presented at the 2007 American Library Association Annual Conference, Washington, D.C., June 2007.

Choi, Y., Hsieh-Yee, I., and Kules, B. (2007, June). *Retrieval Effectiveness of TOC and LCSH.* A paper presented at the Joint Conference on Digital Libraries 2007, Vancouver, Canada.

Vellucci, S. L., Hsieh-Yee, I., and Moen, W.E. (2007, May). *If We Build It, Will They Come? Building a Community of Practice for Metadata Stakeholders.* A poster presented at the Rutgers University Research Day, Bridgeton, New Jersey.

Hsieh-Yee, I. (2007, May). *Federated Searching: User Experience & Perceptions.* International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.

Hsieh-Yee, I. (2007, May). *Search Performance of Google Scholar and Academic Search Premier.* International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.

Hsieh-Yee, I. (2007, May) *MERIC: Building a Digital Commons for Metadata Education & Research.* International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.

Hsieh-Yee, I., and Coogan, J. (2007, March/April). *A Comparative Analysis of Google Scholar and Academic Search Premier.* Poster presented at the Association of College & Research Libraries 13th National Conference, Baltimore, Maryland.

Vellucci, S. L. and Hsieh-Yee, I. (2007, March/April) *They Didn't Teach Me That in Library School! Building a Digital Teaching Commons to Enhance Metadata Teaching, Learning and Research*. On-site presentation and Webcast by Elluminate. A contributed paper presented at the Association of College & Research Libraries 13th National Conference, Baltimore, Maryland. The acceptance rate for contributed paper was 20%. This paper was one of 10 conference papers chosen for live webcast during the conference.

Moen, W., Hsieh-Yee, I. and Vellucci, S.L. (2007, January) *A DSpace Foundation for a Teaching & Research Commons: The Metadata Education and Research Information Commons.* A poster session presented at the Open Repositories Conference 2007, San Antonio, Texas.

Tang, R., Hsieh-Yee, I., and Zhang, S. (2006, November) *User Perception of MetaLib Combined Search*. Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas, Nov. 2006.

Hsieh-Yee, I. (2006, November). *Federated Searching: User Perceptions, System Design, and Library Instructions.* Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas. (Panel organizer, moderator, presenter).

Hsieh-Yee, I. (2006, November). *Building a Digital Teaching Commons to Enhance Teaching and Learning: The MERIC Experience and Challenges.* Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas. (Panel organizer, moderator, presenter)

Hsieh-Yee, I. (2006, September). *Search Performance of Google Scholar and Academic Search Premier.* Paper presented at the ERIC Publishers Meeting, Washington, D.C.

Hsieh-Yee, I., Zhang, S., and Rong Tang, R. (2006, June). *User Perceptions of a Federated Search System*. Poster presented at Joint Conference on Digital Libraries, Chapel Hill, North Carolina.

Hsieh-Yee, I. and Zhang, S. (2006, June). *Preparing Users for Federated Search: Implications of a MetaLib User Perceptions Study.* Paper presented at the 2006 Ex Libris User Groups of North America Conference, Knoxville, Tennessee.

Hsieh-Yee, I. (2006, January). *MERIC Organizations and Navigation*. Paper presented at the 2006 ALISE Annual Conference, San Antonio, Texas.

Hsieh-Yee, I. (2006, January). *Metadata and Cataloging Education: Recommended Competencies.* Paper presented at the 2006 ALISE Annual Conference, San Antonio, Texas.

Hsieh-Yee, I. (2005, November). *Digital Library Evaluation: Progress & Next Steps.* Presentation at the Annual Meeting of the American Society for Information Science & Technology, Charlotte, North Carolina.

Hsieh-Yee, I. (2005, August). *Providing Access to Digital Content: Issues for DL Managers.* Presentation at MDK12 Digital Library Steering Committee Meeting, Columbia, Maryland.

Hsieh-Yee, I. (2005, April). *Enhancing Teaching and Learning: The Role of School Library Media Specialists.* Presentation at Meeting of the Baltimore County Public School System School Media Specialists, Baltimore, Maryland.

Hsieh-Yee, I. (2005, January). *Subject Access and Users: Insights & Inspirations from Marcia J. Bates.* Paper presented at the Historical Perspectives SIG, 2005 Conference of the Association for Library and Information Science Education, Boston, Massachusetts.

Hsieh-Yee, I. (2005, January). *Electronic Resource Management: Practice, Employer Expectations, & CE Interests.* Paper presented at Technical Services Education SIG, 2005 Conference of the Association for Library and Information Science Education, Boston, Massachusetts.

Hsieh-Yee, I. (2004, October). *Library Professionals for the Digital Age: Competencies & Preparation.* Paper presented at Bibliographic Access Management Team meeting, Library of Congress, Washington, D.C.

Hsieh-Yee, I. (2004, January). *Cataloging and metadata expertise for the digital era.* Presented at Preparing 21st Century Cataloging and Metadata Professionals: A Workshop for Educators and Trainers, San Diego and sponsored by ALCTS, ALISE, LC, and OCLC.

Hsieh-Yee, I. (2004, January). *Educating catalogers for the digital era.* Paper presented at the Technical Services SIG, 2004 Conference of the Association for Library and Information Science Education, San Diego.

Hsieh-Yee, I. (2003, July). *Cataloging Education for the 21st Century.* A presentation at the Library of Congress, Washington, D.C.

Hsieh-Yee, I. (2002, January) *Metadata Education and Research Priorities: A Delphi Study of Metadata Experts.* Presentation at the 2002 Conference of the Association for Library and Information Science Education, New Orleans.

Hsieh-Yee, I. (2001, November). *A Delphi Study of Metadata: Preliminary Findings.* Poster session at the 2001 Annual Meeting of the American Society for Information Science & Technology, Washington, D.C.

Hsieh-Yee, I. (2001, June). *Resources on Asian American Children: Analysis of Retrieval by Search Engines and WorldCat.* Presentation at the National Conference on Asian Pacific American Librarians, San Francisco.

Hsieh-Yee, I. (2001, January). *Delphi Study on Metadata: Project Design.* Presentation at Research Awards Session, Association for Library & Information Science Education, Washington, D.C.

Hsieh-Yee, I. (2000, May). *Web Search Behavior Research: Progress and Implications.* Presentation at the Symposium on Evaluating Library and Information Science Research, University of Wisconsin-Madison, Madison, Wisconsin.

Hsieh-Yee, I. (2000, March). *ERIC User Services: Evaluation in a Decentralized Environment.* Presentation at the National ERIC Joint Directors/Technical Meeting, Arlington, Virginia.

Hsieh-Yee, I. (2000, January). *Enhancing Learning with Web Technology.* Presentation at Faculty Conversations, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (2000, January). *From Surrogates to Objects: CUA's Approaches to Organizing Electronic Resources.* Paper presentation at the Annual Conference of the Association for Library and Information Science Education, San Antonio, Texas.

Yee, P., and Hsieh-Yee, I. (1997, November). *Individual Differences in Search Behavior on the WWW.* A poster session presented at the 38th Annual Meeting of the Psychonomic Society, Philadelphia, Pennsylvania.

Hsieh-Yee, I. (1997, April). *Research + Marketing + Preparation = Job!* Presented at the "Workshop on Resume and Interview Techniques," Special Libraries Association, Student Chapter, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (1997, February). *Creating CyberCatalogers: Education and Training.* Presentation at ALA's Midwinter Meeting, Washington, D.C.

Hsieh-Yee, I. (1997, February). *Search Tactics of Web Users in Searching for Texts, Graphics, Known Items and Subjects: A Search Simulation Study.* Presented at the Conference of the Association for Library and Information Science Education, Washington, D.C.

Hsieh-Yee, Ingrid. "Beginning Your Special Library/Information Center Career." Presented at SLA's "Career Day," Jan. 11, 1997, Catholic University of America.

Hsieh-Yee, I. (1996, September). *The Roles of Library and Information Scientists in Managing Electronic Information.* Presentation at Hamilton College, Clinton, New York.

Hsieh-Yee, I. (1996, May). *The Future of Cataloging as a Profession.* Presented at "The Cataloging Forum, Library of Congress, Washington, D.C.

Hsieh-Yee, I. (1994, October). *The Impact of the Internet on OPACs.* Presented at the Third Workshop on User Interfaces for OPACs, Library of Congress, Washington, D.C.

Reports

Hsieh-Yee, I., with Knowledge Management Competencies and Performance Action Group of the Federal Knowledge Management Initiative. "From Knowledge Management Competencies to Improved Organizational Performance." April 9, 2009.

Hsieh-Yee, I., with Knowledge Practices Action Group of the Federal Knowledge Management Initiative. "KM Practice in Government Agencies: Findings and Recommendations." April 9, 2009.

Hsieh-Yee, I. "Delphi Study on Metadata." 2001. Three quarterly reports submitted to the Association for Library and Information Science Education.

Hsieh-Yee, I. "College Students' Information Channels: Patterns of Use and Possible Factors in Channel Selection." 1995. Submitted to the Catholic University of America.

Hsieh-Yee, I. "The Information-Seeking Patterns of Scholars and Their Use of an Online Information System." 1994. Submitted to the Council on Library Resources.

Book Reviews

Review of *The Measurement and Evaluation of Library Services*, by Sharon L. Baker and F. Wilfrid Lancaster. *Information Processing and Management* 30 (1994): 450-52.

Review of *Subject Access to Films and Videos, by Sheila S. Intner and William E. Studwell; and Cataloging Unpublished Nonprint Materials*, by Verna Urbanski with Bao Chu Chang and Bernard L. Karon. *Information Processing and Management* 30 (1994): 449-50.

Review of *Automated Information Retrieval in Libraries: A Management Handbook*, by Vicki Anders. *Journal of Library and Information Science* 19 (1993): 98-100.

Review of *Full Text Databases*, by Carol Tenopir and Jung Soon Ro. *Information Processing and Management* 28 (1992): 667-68.

Review of *Descriptive Cataloging for the AACR2R And USMARC: A How-to-Do It Workbook*, by Larry Millsap and Terry Ellen Ferl. *Information Processing and Management* 28 (1992): 809-11.

Review of *MARC Manual: Understanding and Using MARC Records*, by Deborah J. Byrne. *Information Processing and Management* 28 (1992): 537-38.

Service

Professional Associations and Societies

- Library of Congress. RDA Training Program for the Profession. Co-authored with Tim Carlton. 2013-2014.
- 2014 Digital Preservation Outreach & Education Survey. Contributed to the design of the survey, 2014.
- National Digital Stewardship Alliance. Outreach Committee. 2011-2014.
- National Digital Stewardship Residency Program. Advisory Group, 2012-2013.
- FEDLINK Health Information Technology Advisory Council, 2011-2015.

- 2012 Joint Conference on Digital Libraries. Program Planning Committee, Pre-Conference Proposals Review Committee, 2012
- Catholic Research Resources Alliance. Five-Year Strategic Plan Task Force, 2011-2012
- Institute of Museum and Library Services. Grant reviewer. 2004, 2005, 2010.

- Association for Library and Information Science Education.
  * ALISE Bodan Wynar Research Paper Award Committee, 2015, 2016, 2017
  * ALISE Eugene Garfield Dissertation Award Competition, Jury, 2013, 2014
  * ALISE Research Grant Competition Committee. Chair, 2012
  * Pratt-Severn Faculty Innovation Award. Chair, 2009, 2010
  * ALISE Doctoral Poster Jury, 2012
  * "Information Organization Competencies for the 21st Century" Discussion session leader. 2009 Conference of the Association for Library and Information Science Education.
  * Assisted Technical Services SIG Convener in organizing a program, ""Building a Metadata Education and Research Community through MERIC (Metadata Education and Research Information Commons): Demo and Stakeholder Input" for the 2008 ALISE conference.
  * Association for Library Collections and Technical Services/Association for Library and Information Science Education (ALCTS/ALISE) Metadata Education and Research Information Center (MERIC) Advisory Board, Co-Chair (with Sherry Vellucci), 2005-2007. Chair, 2008-2009 (leading the effort to build MERIC, a repository and collaborative space for metadata educators, practitioners, and researchers)
  * Technical Services SIG, Convener, 2004-2005. Organized a program on "Electronic Resources Management: Current Practices, Employer Expectations, and Teaching Strategies" for the 2005 conference in Boston, Massachusetts.
  * Technical Services SIG, Convener, 2003-2004. Organized a program on "Organizing Information with Metadata: Desired Competencies and Teaching Innovations" for the 2004 conference.
  * Technical Services SIG, Convener, 1999-2000. Organized a program on "Teaching the Organization of Electronic Resources" for the 2000 conference.
  * Curriculum SIG, Co-convener (with Sibyl Moses), 1996-97. Organized a program on "Government Information Policy" for the 1997 conference.

- American Society for Information Science & Technology.
  * Reviewer, Conference program panel submissions and poster submissions, 2005, 2006, 2007, 2009, 2011, 2012, 2013, 2014, 2015, 2016, 2017
  * Nomination Committee, 2009-2011
  * Information Science Education Special Interest Group. American Society for Information Science and Technology. Chair-Elect, 2007-2008. Chair 2008-2009.
  * Committee on Information Science Education. 1999-2006.
  * Committee on Information Science Education. Organizing Committee for an orientation program for students at ASIS annual meetings, 1999-2001
  * Committee on Information Science Education. Sub-committee on Student Welfare (focusing on issues related to master's education), 1998-2001

* SIG ED. Organizing Committee for the "Seminar on Research and Career Development" for junior researchers. 1995-96 (chair), 1997-2001
* ISI Doctoral Dissertation Proposal Scholarship Jury, 1997; 2001, 2002
* Pratt-Severn Best Student Research Paper Award Jury. Chair. 1997
* 1998 Midyear Meeting (referee of contributed papers), 1997
* Organizer and moderator of the ASIS Doctoral Forum and the Doctoral Research Seminar 1994-1995
* SIG Human Computer Interaction. Chair-Elect, Chair, 1993-1995
* Doctoral Forum Award Jury, 1995
* Best Student Paper Award Jury, 1995

- American Library Association.
  * Committee on Accreditation, External Review Panelist, 2009- (site visiting team 2013-2014; site visiting team 2016-2017)
  * Association for Library Collections and Technical Services Task Force on Competencies and Education for a Career in Cataloging, member, 2008-2009
  * Facilitator for "What They Don't Teach in Library School: Competencies, Education and Employer Expectations for a Career in Cataloging," an Association for Library Collections and Technical Services Preconference, June 22, 2007 in Washington, D.C. Also a local liaison for bringing this program to the Catholic University of America.
  * Facilitator for a discussion on "Effect of Electronic Resources on Technical Services" at ALA's Midwinter Meeting held in Feb. 1997 in Washington, D.C.
  * International Relations Committee, Subcommittee Task Force for IFLA and China, 1994-1997

- Virginia Association of School Librarians. Scholarships and Awards Committee. 2010-2012

- Federal Knowledge Management Initiative, Knowledge Management Practices Action Group. Member. 2009 (leading the effort to build a knowledge management repository)
- Federal Knowledge Management Initiative, Knowledge Management Competencies & Learning Action Group. Member. 2009 (developing an action plan for helping government knowledge workers and government agencies to develop knowledge management competencies)
- National Center for Education Statistics. Technical Review Panel. 2008.
- External evaluator for a case of promotion to full professorship. University of Tennessee. 2008.
- National Information Standards Organization (NISO). Advisory Board, Revision of "IMLS Framework of Guidance for Building Good Digital Collections," 2004, 2007.
- Library of Congress, Bibliographic Control of Web Resources: A Library of Congress Action Plan. Principal Investigator of Action Item 5.1, focusing on cataloging and metadata education for students and new librarians, 2002-2003. (worked with the Association for Library Collections and Technical Services, Education Task Force)

- Chinese American Librarians Association
  * Chinese American Librarians Association Outstanding Library Leadership Award in Memory of Dr. Margaret Chang Fung, Award Committee, 2016-2017
  * Achievement Award Jury, 2000-2001
  * CALA Goal 2000 Task Force, 1997
  * Scholarship Committee, 1995, 1996-1997 (chair)
  * Board of Directors, 1994-1997
  * Publication Committee, 1993-1995
  * International Relations Committee, 1993-1996

- Sailor[SM] Assessment Advisory Group (An impact study of Sailor, Maryland's Public Information Network), 1995

- Editorial boards
  Journal of Library & Information Science. Editorial Board, 2012-
  Chinese American Librarians Association, *Occasional Papers Series*. Editorial Board, 2009-2016.
  Library Quarterly. Editorial Board, 2003-2008
  Bulletin of the Medical Library Association, 1994-97
  Newsletter editor for the Chinese American Librarians Association, 1989-92

- Referee for the following journals
  Information Processing and Management
  Journal of Digital Information
  Journal of Education for Library and Information Science
  Journal of Information Science
  Journal of Library & Information Science
  Journal of Library Metadata
  Journal of the American Society for Information Science & Technology
  Library and Information Science Research
  Library Quarterly

- Expert reviewer, "Digital Library" course, Evaluation module, University of North Carolina, Chapel Hill, 2007-2008.
- Expert reviewer, "Information Organization" course, University of Michigan, Ann Arbor. 2007.

Catholic University of America

- Academic Senate Committee on Appointments and Promotions, 2005-2008
- Academic Senate Committee on Committees and Rules, 2009-2012, 2019-2022
- Academic Senate representative (for School of Arts & Sciences), 2017-2020

- Academic Senate representative (for School of Library & Information Science) 2003-2012
- Academic Senate Library Committee, Interim Chair (2007), Member, 2008-2012
- Academic Senate Committee on Computing, 1995-2003

- President's Administrative Council, 2010-2012
- Deans' Council, 2010-2012
- Academic Leadership Group, 2010-2012
- Graduate Board, 2010-2012

- School of Arts & Sciences, Committee on Appointments and Promotions, 2015-2019
- School of Arts & Sciences, Academic Council, 2015-2016.
- School of Arts & Sciences, Ordinary Professor Group, 2013-
- Doctoral Dissertation Defense Committee, Chair, Dept. of Psychology, 2016, 2017, 2018, 2019, 2020, 2021, 2022
- Doctoral Dissertation Defense Committee, Chair, Dept. of Education, 2014, 2015, 2017, 2018, 2019
- Doctoral Dissertation Defense Committee, Chair, National Catholic School of Social Services, 2019, 2020, 2021
- Doctoral Dissertation Defense Committee, Chair, School of Nursing, 2006, 2008
- CUA Scholarly Communication Project Team, Member (2007), Chair, 2008-2009
- Dean Search Committee, 1992-1994, 1998-1999, 2002-2003, 2006-2007
- Fulbright Review Panel, 2006
- CUA Service Learning Advisory Board, 2001-2002
- CUA Faculty Conversations on Enhancing Teaching and Learning through Technology, Planning Group, 1999-2001
- CUA Initiative on Technology and Teaching, 1998-2001

Dept. of Library and Information Science

- Admissions Committee, 2007-2009, Chair 2010-2012, Member 2013-2015, Member 2018-2022
- Appointments and Promotions Committee, 1991-2022
- Student advisement, 1990-2022
- 4+1 Bachelor to Master program, Program Advisor, 2014-2022
- LIS Minor, Advisor, 2017-2022
- Curriculum Committee, 2020-21 (Member), 2021-2022 (Chair)
- Scholarship & Awards Committee, 2016-2018, 2020-21
- Comprehensive examination editor, 2020-21
- Symposium, Colloquium, Lecture Series Committee, May 2018-May 2020 (Chair)
- Symposium and Colloquium Committee, fall 2016-May 2018
- Bridging the Spectrum Symposium session moderator, 2018, 2019, 2020, 2021

- Organization of Information Core Course Outcomes Assessment, 2019, 2020
- The Information Professions in Society Core Course Outcomes Assessment, 2019, 2020
- Community Services Librarianship, Course of Study Review and Revision, 2020
- Organization of Information, Course of Study Review and Revision, 2019
- Organization of Information, Competency and Course Offerings Review, 2018-2019
- Accreditation presentation, Chair, June 2015-August 2016
- Interim Co-Chair, June 2015-August 2016.
- Blended/OWL Learning Committee, spring 2016-2018
- Technology Committee, fall 2016-2017
- Comprehensive examination editor, 2016-2017, reader (every year since 1990)
- LIS Advisory Board, 2015-2016 (chair); fall 2016- May 2018 (member)
- Committee on Planning and Assessment, 2015-2016 (chair)
- Senior Faculty Committee, 2014-2016.
- Accreditation Steering Committee, 2014-2016 (Chair, 2015-2016)
- Accreditation Students Standard Committee, co-chair, 2014-2016
- Accreditation Mission, Goals, and Objectives Standard Committee, co-chair, 2014-2016
- Accreditation Curriculum Standard, member 2014-2-16
- Accreditation Administration and Finance Standard, member 2014-2016
- Cultural Heritage Information Management Project (IMLS-funded), Co-PI, 2012-2015
- Cultural Heritage Information Management Forum (scheduled for June 2015), Co-Organizer, 2013-2015
- Health Information Technology Interim Review Committee, 2015 (chair)
- Health Sciences Librarianship Advisory Group, 2015- (chair)
- Comprehensive examination editors, 2013-2014, 2016-2017
- National Digital Stewardship Alliance liaison, 2011-2014
- Advisory Board, Chair 2010-2012
- Academic Honesty Committee, Chair, 2008-2012
- Blended Learning Committee, 2010-2012
- Colloquium Committee, 2010-2012
- Comprehensive Examination Administration, 2010-2012
- Cultural Heritage Information Management Advisory Committee, 2010-2012 (chair), 2013-
- Curriculum Committee, 1991-2003, 2007-2009, Chair 2010-2012, member 2013-
- Curriculum Subcommittee on Comprehensive Examination, Chair 2009-2012
- Health Information Technology Advisory Board, Chair 2010-2012. Member 2013-
- Health Sciences Advisory Committee, 2009, Chair 2010-2012. Member 2013-
- HIT Expert Forum, Chair 2012. Member 2013-2015
- Health Information Technology Student Group Advisor, 2011-2012
- State Council for Higher Education of Virginia, SLIS Representative, 2010-2012
- Symposium Planning Committee, 2010-2012
- Website Management Team, Chair, 2010-2012
- Urban School Librarianship Project (IMLS-Funded), PI, 2007-2011 (chair, 2010-11)

- Failing Grades Committee, 1995-1997 (chair), 2000-2001 (chair), 2004-2005 (chair), 2007 (chair)-2011
- Faculty Search Committee, 1994-1998, 2002-2004, 2006 (chair), Fall 2007-2009, Chair fall 2009-2012
- Recruitment Committee, Chair 2010-2012
- Strategic Planning Committee, Chair 2010-2012
- Technology Committee, 2010-2012
- Accreditation Advisory Committee, 2007-2009
- Accreditation Coordinating Committee, 2007-2009
- Accreditation Steering Committee, 2007-2009
- SLIS Advisory Group, 2007-2009
- Accreditation Curriculum Standard Committee, Co-chair, 2007-2009
- Accreditation Faculty Standard Committee, Co-chair, 2007-2009
- LSC 551 Information Organization Review Team, Co-chair, 2008-2009, 2015-2016.
- Curriculum Subcommittee on Portfolios, 2009
- LSC 555 Information Systems in Libraries and Information Centers Review Team, contributor, 2008-2009
- Redesign of LSC 730 Use and Users of Libraries and Information. 2009-
- Development of a metadata institute that was taught as LSC 715 Organization of Internet Resources in 2008. The institute is being revised and will be offered in 2010 under a new course title.
- Development of lesson plans, assignments, and evaluation rubrics for LSC 606, Cataloging and Classification, for the School's NCATE accreditation. 2008
- Howard and Mathilde Rovelstad Scholarship Committee, Chair, 2004-2007
- Assistant Dean Search Committee, Chair, Fall 2007
- Liaison to the Association for Library Collections and Technical Services to bring its preconference program, Cataloging Education and Employer Expectations, to CUA during the 2007 American Library Association Annual Meeting in Washington, D.C.
- Organizer of the colloquium presentation and reception for Tamar Sadeh of Ex Libris on PRIMO June 2007
- Practicum review and design (work with potential supervisors, such as the American Indian Museum internship description revision) 2006-
- Comprehensive examinations (edits, proctoring, and grading), 1990-
- SLIS Web site redesign: Comments and suggestions. Fall 2007
- Conducted surveys of current students and alumni in preparation for the 2005 re-accreditation, 2004-2005
- Technology Committee, 1992-1999 (chair, 1996-1998), 2002-2003 (member)
- Colloquia Committee 1997-1999, 2002-2003.
- Advisor of the CUA Student Chapter of the American Society for Information Science and Technology, 2002-2003
- Visiting Professor Search Committee, 1999, 2000, 2001
- Leader, Participation in the CORC experiment, 1999-2000
- Advisor of the Special Libraries Association Student Chapter, 1993-1999; the group was recognized for outstanding leadership by SLA in 1999.

- COA planning Committee, Task Force on Electronic Presentation of SLIS Reports (team leader) 1997-1998
- COA Planning Committee, Subcommittee on Technology 1996-1998
- NLM practicum coordinator, 1997-1998
- Computer Literacy Workshops: Assisted with the development and evaluation of the workshops, 1996-1998
- Leader, Participation in the InterCat project, 1995-1997

Appendix B

MARC 21 Reference Materials

Part VII: A Summary of Commonly Used MARC 21 Fields
Part VIII: A List of Other Fields Often Seen in MARC Records
Part IX: The Leader
Part X: Field 008 for Books

Part VII:

A Summary of Commonly Used MARC 21 Fields

This is a summary of the MARC 21 tags used most frequently by libraries in entering their own bibliographic records. For full listings of all MARC 21 tags, indicators, and subfield codes, see *MARC 21 Format for Bibliographic Data*.

In the explanations on these pages:

Tags -- The tags (3-digit numbers) are followed by the names of the fields they represent. In this summary, and in the *MARC 21 Format for Bibliographic Data*, if a tag can appear more than once in one bibliographic record, it is labeled repeatable (R). If it can only be used once, it is labeled non-repeatable (NR). For example, a catalog record can have several subjects, so the tags for subject added entries (6XX) are labeled repeatable (R).

Indicators -- The use of indicators is explained in fields where they are used. Indicators are one-digit numbers. Beginning with the 010 field, in every field -- following the tag -- are two character positions, one for Indicator 1 and one for Indicator 2. The indicators are not actually defined in all fields, however. And it is possible that a 2nd indicator will be used, while the 1st indicator remains undefined (or vice versa). When an indicator is undefined, the character position will be represented by the character # (for blank space).

Subfield codes -- All the data in each field (beginning with the 010 field) is divided into subfields, each of which is preceded by a delimiter-subfield code combination. The most common subfield codes used with each tag are shown. Each subfield code is preceded by the character $, signifying a delimiter. The name of the subfield follows the code.

In general, every field MUST have a subfield 'a' (**$a**). One exception that is often seen is in Field 020 (ISBN), when the ISBN information (subfield **$a**) is unavailable but the price (subfield **$c**) is known. Some subfields are repeatable. In this summary, repeatability is noted for only the more common repeatable subfields.

Examples: Examples follow the explanation for each field. For clarity, one space has been placed between the tag and the first indicator, one space has been placed between the second indicator and the first delimiter- subfield code, and one space has been inserted between the delimiter-subfield code and the subfield data.

Google Exhibit No. 1063
Page 0174

010 Library of Congress Control Number -- (LCCN)
(NR, or Not Repeatable
Indicators undefined.
Subfield used most often:
$a -- Library of Congress control number

```
Example:    010 ##  $a ###86000988#
```

020 International Standard Book Number -- (ISBN)
(R, or Repeatable)
Indicators undefined.
Subfields used most often:
$a -- International Standard Book Number
$c -- Terms of availability (often a price)
$z -- Cancelled/invalid ISBN (R)

```
Example:    020 ##  $a 0877547637
```

040 Cataloging source -- (NR)
Indicators undefined.
Subfields used most often:
$a -- Original cataloging agency
$c -- Transcribing agency
$d -- Modifying agency (R)

```
Example:    040 ## $a DLC
                   $c DLC
                   $d gwhs
```

100 Main entry -- Personal name -- (primary author)
(NR; there can be only one main entry)
Indicator 1: Type of personal name entry element
0 -- Forename
1 -- Surname (this is the most common form)
3 -- Family name
Indicator 2 undefined.
Indicator 2 became obsolete in 1990. Older records may display 0 or 1
Subfields used most often:
$a -- Personal name
$b -- Numeration

Google Exhibit No. 1063
Page 0175

$c -- Titles and other words associated with a name (R)
$q -- Fuller form of name
$d -- Dates associated with a name (generally, year of birth)

```
Example:    100 1# $a Gregory, Ruth W.
                   $q (Ruth Wilhelme),
                   $d 1910–
```

130 Main entry -- Uniform title -- (NR)

Indicator 1: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Indicator 2 undefined.

Indicator 2 became obsolete in 1990. (See 100 above.)

Subfields used most often:

$a -- Uniform title
$p -- Name of part/section of a work (R)
$l -- Language of a work
$s -- Version
$f -- Date of a work

```
Example:    130 0# $a Bible.
                   $p O.T.
                   $p Psalms.
```

240 Uniform title (NR)

Indicator 1: Uniform title printed or displayed

0 -- Not printed or displayed
1 -- Printed or displayed (most common)

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Subfields used most often:

$a -- Uniform title
$l -- Language of a work
$f -- Date of a work

```
Example:    240 10 $a Ile mystérieuse.
                   $l English.
                   $f 1978
```


Google Exhibit No. 1063
Page 0176

245 Title Statement **(NR)**

Indicator 1: Title added entry

(Should the title be indexed as a title added entry?)

0 -- No title added entry

(indicates a title main entry; i.e. no author is given)

1 -- Title added entry

(the proper indicator when an author given in 1XX; the most common situation)

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present, including spaces; usually set at zero, except when the title begins with an article; e.g., for *The robe*, the second indicator would be set to 4. The letters *T, h, e,* and the space following them are then ignored in alphabetizing titles. The record will be automatically filed under "*r*" -- for *Robe*.

Subfields used most often:

$a -- Title proper

$h -- Medium (often used for non-book media)

$p -- Name of part/section of a work (R)

$b -- Reminder of title (subtitles, etc.)

$c -- Remainder of title page transcription/Statement of responsibility

```
Example:    245 14 $a The DNA story :
                   $b a documentary history of gene
                      cloning /
                   $c James D. Watson, John Tooze.
```

246 Varying form of title (R)

Indicator 1: Note/title added entry controller

1 -- Note, title added entry

3 -- No note, title added entry

Indicator 2: Type of title

-- No information provided

0 -- Portion of title

1 -- Parallel title

4 -- Cover title

8 -- Spine title

Subfield used most often:

$a -- Title proper

```
Example:    246 3# $a Four corners power review
```

250 Edition statement (NR)

Google Exhibit No. 1063
Page 0177

Indicators undefined.
Subfield used most often:
$a -- Edition statement

```
Example:    250 ## $a 6th ed.
```

260 Publication, distribution, etc. (Imprint) (R)
Indicator 1: Sequence of publishing statements
-- No information provided
Indicator 2: Undefined
Subfields used most often:
$a -- Place of publication, distribution, etc. (R)
$b -- Name of publisher, distributor, etc. (R)
$c -- Date of publication, distribution, etc. (R)

```
Example:    260 ## $a New York :
                   $b Chelsea House,
                   $c 1986.
```

300 Physical description (R)
Indicators undefined.
Subfields used most often:
$a -- Extent (number of pages) (R)
$b -- Other physical details (usually illustration information)
$c -- Dimensions (cm.) (R)
$e -- Accompanying material (for example, "teacher's guide" or "manual")

```
Example:    300 ## $a 139 p. :
                   $b ill. ;
                   $c 24 cm.
```

440 Series statement / Added entry--Title
This field was made obsolete in 2008 to simplify the series statement. See 490 and 830.

490 Series statement (No added entry is traced from field) (R)
Indicator 1: Specifies whether series is traced (whether an 8XX tag is also present)
0 -- Series not traced
1 -- Series traced (8XX is in record)
Indicator 2 undefined.
Subfield used most often:

Google Exhibit No. 1063
Page 0178

$a -- Series statement (R)
$v -- Volume number (R)

```
Example:    490 1# $a Colonial American craftsmen
```

500 General note (R)

Indicators undefined.

Subfield used most often:

$a -- General note (Used when no specialized note field has been defined for the information. Examples: Notes regarding the index; the source of the title; variations in title; descriptions of the nature, form, or scope of the item.)

```
Example:    500 ##  $a Includes index.
```

504 Bibliography, etc. note (R)

Indicators undefined.

Subfield used most often:

$a -- Bibliography, etc. note

```
Example:    504 ## $a Includes bibliographical
                      references.
```

505 Formatted contents note (R)

Indicator 1: Type of contents note

0 -- Complete contents
1 -- Incomplete contents (used with multivolume set when some volumes are not yet published)
2 -- Partial contents

Indicator 2: Level of content designation

-- Basic

Subfield used most often:

$a -- Formatted contents note

```
Example:    505 0# $a Pride and prejudice -- Emma
                      -- Northanger Abbey.
```

520 Summary, etc. note (R)

Indicator 1: Display constant controller

-- Summary

Google Exhibit No. 1063
Page 0179

1 -- Review
2 -- Scope and content
3 -- Abstract

Indicator 2 undefined

Subfields used most often

$a -- Summary, abstract, or annotation
$b -- Expansion of summary note

```
Example:    520 ## $a This basic guide to parliamentary
                     procedure tells how to conduct
                     and participate in a meeting
                     properly.
```

600 Subject added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

0 -- Forename
1 -- Surname (this is the most common form)
3 -- Family name

Indicator 2: Subject heading system/thesaurus (identifies the specific list or file which was used)

0 -- Library of Congress Subject Headings
1 -- LC subject headings for children's literature
2 -- Medical Subject Headings
3 -- National Agricultural Library subject authority file
4 -- Source not specified
5 -- Canadian Subject Headings
6 -- Répertoire de vedettes-matière
7 -- Source specified in subfield $2

(Note regarding Sears subject headings: The MARC 21 format does not provide an assigned indicator for Sears subject headings. Therefore, an indicator of 7 is used, and the MARC defined code "sears" is placed in subfield $2.)

Subfields used most often:

$a -- Personal name (surname and forename)
$b -- Numeration
$c -- Titles and other words associated with a name (R)
$q -- Fuller form of name
$d -- Dates associated with a name (generally, year of birth)
$t -- Title of a work
$v -- Form subdivision (R)
$x -- General subdivision (R)
$y -- Chronological subdivision (R)
$z -- Geographic subdivision (R)
$2 -- Source of heading or term (used with 2nd indicator of 7)

Google Exhibit No. 1063
Page 0180

```
Example:    600 10 $a Shakespeare, William,
                   $d 1564-1616
                   $x Comedies
                   $x Stage history.
```

```
Example:    600 10 $a Shakespeare, William,
                   $d 1564-1616
                   $x Knowledge
                   $z Rome
                   $v Congresses.
```

Notice that subfields $v, $x, and $z in the 600 field are repeatable. Subfields $v, $x, $y, and $z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

610 Subject added entry -- Corporate name (R)

*Indicator 1:*Type of corporate name entry element

0 -- Inverted name (not used with AACR2)

1 -- Jurisdiction name

2 -- Name in direct order

Indicator 2: Subject heading system/thesaurus.

See indicator 2 under 600

Subfields used most often:

$a -- Corporate name or jurisdiction name as entry element

$b -- Subordinate unit (R)

$v -- Form subdivision (R)

$x -- General subdivision (R)

$y -- Chronological subdivision (R)

$z -- Geographic subdivision (R)

$2 -- Source of heading or term (used with 2nd indicator of 7)

```
Example:    610 10 $a United States.
                   $b Army Air Forces
                   $v Biography.
```

650 Subject added entry -- Topical term (Most subject headings fit here.) (R)

Indicator 1: Level of subject

-- No information provided

Indicator 2: Subject heading system/thesaurus
(identifies the specific list or file which was used)

0 -- Library of Congress Subject Headings

1 -- LC subject headings for children's literature

Google Exhibit No. 1063
Page 0181

2 -- Medical Subject Headings
3 -- National Agricultural Library subject authority file
4 -- Source not specified
5 -- Canadian Subject Headings
6 -- Répertoire de vedettes-matière
7 -- Source specified in subfield $2

Note regarding Sears subject headings: The MARC 21 format does not provide an assigned indicator for Sears subject headings. Therefore, an indicator of 7 is used, and the MARC defined code "sears" is placed in subfield $2.)

Subfields used most often:
$a -- Topical term
$v -- Form subdivision (R)
$x -- General subdivision (R)
$y -- Chronological subdivision (R)
$z -- Geographic subdivision (R)
$2 -- Source of heading or term used with 2nd indicator of 7)

```
Example:    650 #0 $a Theater
                   $z United States
                   $v Biography
                   $v Dictionaries.
```

Notice that subfields $v, $x, and $z in the 650 field are repeatable. Subfields $v, $x, $y, and $z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

651 Subject added entry -- Geographic name (R)
Indicator 1: undefined.
Indicator 2: Subject heading system/thesaurus.
See indicator 2 under 600
Subfields used most often:
$a -- Geographic name
$v -- Form subdivision (R)
$x -- General subdivision (R)
$y -- Chronological subdivision (R)
$z -- Geographic subdivision (R)
$2 -- Source of heading or term (used with 2nd indicator of 7)

```
Example:    651 #0 $a United States
                   $x History
                   $v Chronology.
```


Google Exhibit No. 1063
Page 0182

Notice that subfields $v, $x, and $z in the 651 field are repeatable. Subfields $v, $x, $y, and $z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

700 Added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

0 -- Forename
1 -- Surname (this is the most common form)
3 -- Family name

Indicator 2: Type of added entry

-- No information provided (most common; co-authors, editors, etc.)
2 -- Analytical entry (The values for Indicator 2 changed in 1994 with Format Integration, and older records may display additional values. An analytical entry involves an author/title of an item contained in a work.)

Subfields used most often:

$a -- Personal name
$b -- Numeration
$c -- Titles and other words associated with a name (R)
$q -- Fuller form of name
$d -- Dates associated with a name (generally, year of birth)
$e -- Relator term (such as ill.) (R)
$4 -- Relator code (R)

```
Example:    700 1# $a Baldridge, Letitia.
```

710 Added entry -- Corporate name (R)

Indicator 1: Type of corporate name entry element

0 -- Inverted name (not used with AACR2)
1 -- Jurisdiction name
2 -- Name in direct order

Indicator 2: Type of added entry.

See Indicator 2 under 700
-- No information provided
2 -- Analytical entry

Subfields used most often:

$a -- Corporate name or jurisdiction name as entry element
$b -- Subordinate unit (R)

```
Example:    710 2# $a Sunburst Communications (Firm)
```

740 Added entry -- Uncontrolled related/analytical title (R)

Indicator 1: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including

Google Exhibit No. 1063
Page 0183

spaces)

*Indicator 2:*Type of added entry. See Indicator 2 under 700

-- No information provided

2 -- Analytical entry

(This field was redefined in 1994 with Format Integration. Prior to 1994, the field was also used for variant titles, such as a different wording on a spine title. In records created since Format Integration, those variant tiles appear in a 246 field.)

Subfield used most often:

$a -- Title

```
Example:    740 02 $a Uncle Vanya.
```

800 Series added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

0 -- Forename

1 -- Surname

3 -- Family name

Indicator 2 undefined.

Subfields used most often:

$a -- Personal name

$b -- Numeration

$c -- Titles and other words associated with a name (R)

$q -- Fuller form of name

$d -- Dates associated with a name (generally, year of birth)

$t -- Title of a work (the series)

$v -- Volume number

```
Example:    800 1# $a Fisher, Leonard Everett.
                   $t Colonial American craftsmen.
```

830 Series added entry -- Uniform title (R)

Indicator 1 undefined.

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Subfield used most often:

$a -- Uniform title

$v -- Volume number

```
Example:    830 #0 $a Railroads of America (Macmillan)
```


Google Exhibit No. 1063
Page 0184

[Back to Top of Page]

Part VIII:

A List of Other Fields Often Seen in MARC Records

001	Control number
003	Control number identifier
005	Date and time of latest transaction
006	Fixed-length data elements -- additional material characteristics
007	Physical description fixed field
008	Fixed length data elements (See Part X)
022	International Standard Serial Number (ISSN)
037	Source of acquisition
041	Language code
043	Geographic area code
050	Library of Congress call number
060	National Library of Medicine call number
082	Dewey Decimal classification number (the one recommended by the Library of Congress; locally-assigned call numbers may appear elsewhere)
110	Main entry -- Corporate name (less frequent under AACR2 rules)
256	Computer file characteristics
263	Projected publication date (indicates a CIP -- Cataloging in Publication -- record)
306	Playing time
508	Creation/production credits note
510	Citation/references note (review sources)
511	Participant or performer note
521	Target audience note (first indicator: 0 = reading grade level, 1 = interest age level, 2 = interest grade level, 3 = special audience characteristics, 4 = motivation interest level)
530	Additional physical form available note
538	System details note
586	Awards note
656	Index term -- Occupation
730	Added entry -- Uniform title
852	Location
856	Electronic location and access
9XX	Reserved for local use. (They are used by vendors, systems, or individual libraries to exchange additional data)

[Back to Top of Page]

Google Exhibit No. 1063
Page 0185

Part IX:

The Leader

There are 24 positions in the Leader, numbered from 00 to 23. For fuller explanation, see the *MARC 21 Format for Bibliographic Data*.

00-04 Record length (calculated by the computer for each record)
05 Record status
- a = increase in encoding level
- c = corrected or revised
- d = deleted
- n = new
- p = increase in encoding from prepublication (previous CIP)

06 Type of record
- a = language material
- c = printed music
- d = manuscript music
- e = cartographic material
- f = manuscript cartographic material
- g = projected medium
- i = nonmusical sound recording
- j = musical sound recording
- k = 2-dimensional nonprojectable graphic
- m = computer file
- o = kit
- p = mixed materials
- r = 3-dimensional artifact or naturally occurring object
- t = manuscript language material

07 Bibliographic level
- a = monographic component part
- b = serial component part
- c = collection
- d = subunit
- i = integrating resource
- m = monograph/item
- s = serial

08 Type of control
- # = no specified type
- a = archival

09 Character coding scheme
- # = MARC-8
- a = UCS/Unicode

10 Indicator count (always "2")
11 Subfield code count (always "2")
12-16 Base address of data (calculated by the computer for each record)
17 Encoding level
- # = full level

Google Exhibit No. 1063
Page 0186

1 = full level, material not examined
2 = less-than-full level, material not examined
3 = abbreviated level
4 = core level
5 = partial (preliminary) level
7 = minimal level
8 = prepublication level (CIP)
u = unknown
z = not applicable

18 Descriptive cataloging form
= non-ISBD
a = AACR2
i = ISBD
u = unknown

19 Multipart resource record level
= Not specified or not applicable
a = Set
b = Part with independent title
c = Part with dependent title

20 Length of the length-of-field portion (always "4")
21 Length of the starting-character-position portion (always "5")
22 Length of the implementation-defined portion (always "0")
23 Undefined (always "0")

[Back to Top of Page]

Part X:

Field 008 for Books

Field 008 is used for Fixed Length Data Elements ("Fixed Field Codes"). There are 40 character positions in field 008, numbered from 00-39. Undefined positions must contain either a blank (#) or a fill character (|). Positions 00-17 and 35-39 are defined the same way for all media.

The information shown here for positions 18-34 applies only to books. For explanation of all the positions below and for positions 18-34 for other media, see the *MARC 21 Format for Bibliographic Data*.

Note that field 008 has no indicators or subfield codes.

00-05 Date entered on file (YYMMDD),
where Y=year, M=month, and D=day

06 Type of date/publication status:
b = no dates given; B.C. date involved
e = detailed date
s = single known date/probable date
m = multiple dates
r = reprint/reissue date (Date 1) and original date (Date 2)

Google Exhibit No. 1063
Page 0187

n = dates unknown
q = questionable date
t = publication date and copyright date
| = no attempt to code

07-10 Date 1/beginning date of publication

11-14 Date 2/ending date of publication

Date fields contain the year(s) of publication. The type of date(s) in these elements are specified in fixed field element 06: Type of date/publication status. (For further details, see the field 008 description in the *MARC 21 Format for Bibliographic Data*.)

15-17 Place of publication, production, or execution

For example:
pk# = Pakistan
cau = California (US)

(For a full list of codes used in these positions, see the *MARC Code List for Countries*.)

18-21 Illustrations (up to 4 codes):

= no illustrations
a = illustrations
b = maps
c = portraits
d = charts
e = plans
f = plates
g = music
h = facsimiles
i = coats of arms
j = genealogical tables
k = forms
l = samples
m = phonodisc, phonowire, etc.
o = photographs
p = illuminations
| = no attempt to code

22 Target audience:

= unknown or not specified
a = preschool
b = primary
c = pre-adolescent
d = adolescent
e = adult
f = specialized
g = general
j = juvenile
| = no attempt to code

23 Form of item:

= none of the following
a = microfilm

Google Exhibit No. 1063
Page 0188

b = microfiche
c = microopaque
d = large print
f = braille
r = regular print reproduction
s = electronic
| = no attempt to code

24-27 Nature of contents (up to 4):
= no specified nature of contents
a = abstracts/summaries
b = bibliographies (is one or contains one)
c = catalogs
d = dictionaries
e = encyclopedias
f = handbooks
g = legal articles
i = indexes
j = patent document
k = discographies
l = legislation
m = theses
n = surveys of literature
o = reviews
p = programmed texts
q = filmographies
r = directories
s = statistics
t = technical reports
u = standards/specifications
v = legal cases and notes
w = law reports and digests
z = treaties
| = no attempt to code

28 Government publication:
= not a government publication
i = international intergovernmental
f = federal/national
a = autonomous or semi-autonomous component
s = state, provincial, territorial, dependent, etc.
m = multistate
c = multilocal
l = local
z = other type of government publication
o = government publication -- level undetermined
u = unknown if item is government publication
| = no attempt to code

29 Conference publication:
0 = not a conference publication
1 = conference publication

Google Exhibit No. 1063
Page 0189

| = no attempt to code

30 Festschrift:
0 = not a festschrift
1 = festschrift
| = no attempt to code

31 Index:
0 = no index
1 = index present
| = no attempt to code

32 Undefined (since 1990) (Earlier records may contain the values 0 or 1)
= Undefined
| = no attempt to code

33 Literary form:
0 = not fiction (not further specified)
1 = fiction (not further specified)
c = comic strips
d = dramas
e = essays
f = novels
h = humor, satires, etc.
i = letters
j = short stories
m = mixed forms
p = poetry
s = speeches
u = unknown
| = no attempt to code

34 Biography:
= no biographical material
a = autobiography
b = individual biography
c = collective biography
d = contains biographical information
| = no attempt to code

35-37 Language:
A three-letter code. For example: eng fre ger spa rus ita

(For a full list of codes used in these positions, see the *MARC Code List for Languages*.)

38 Modified record:
= not modified
x = missing characters (because of characters unavailable in MARC character set)
s = shortened
d = "dashed-on" information omitted
r = completely romanized/printed cards in script
o = completely romanized/printed cards romanized
| = no attempt to code

39 Cataloging source:

Google Exhibit No. 1063
Page 0190

= national bibliographic agency
c = cooperative cataloging program
d = other sources
u = unknown
| = no attempt to code

[Back to Top of Page]

[Back to Table of Contents] -- [Continue to Part 11]



Library of Congress
Library of Congress Help Desk (10/27/2009)

Google Exhibit No. 1063
Page 0191

Appendix C

Appendix C. Materials Considered

Main exhibits and Supporting documents

(Supporting documents/materials include a cover sheet on the associated exhibit number and (outside of Appendices A & B) have been appended as part of Appendix C)

Exhibit	Description
1004	Mattavelli, M., & Brünettem, S. (1998, September). Computational graceful degradation for video decoding. In *Signal Processing IX: Theories and Applications : Proceedings of EUSIPCO-1998*, pp. 1-4 ("*Mattavelli*").
1004A	Copy of *Mattavelli* from the EUSIPCO 1998 conference proceedings, available at https://eurasip.org/Proceedings/Eusipco/Eusipco1998/sessions/F%20A/FA%20S-3/2309/694.PDF.
1004B	Bibliographic and MARC records for *Signal Processing IX: Theories and Applications : Proceedings of EUSIPCO-1998* that contains *Mattavelli*, available at https://lccn.loc.gov/98232284 from the online catalog of the Library of Congress.
1004C	Profile of Library of Congress, available at https://librarytechnology.org/library/2263
1007	Wang, S. H., Peng, W. H., He, Y., Lin, G. Y., Lin, C. Y., Chang, S. C., ... & Chiang, T. (2003, December). A platform-based MPEG-4 advanced video coding (AVC) decoder with block level pipelining. In *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*, Vol. 1, pp. 51-55 ("*Wang*").
1007A	Copy of *Wang* obtained from IEEE.
1007B	IEEE metadata record for *Wang*, available at https://ieeexplore.ieee.org/abstract/document/1292411 from IEEE Xplore.
1007C	IEEE Xplore archived on February 7, 2001, by Internet Archive, available at https://web.archive.org/web/20010207003457/http://ieeexplore.ieee.org/lpdocs/epic03/
1007D	Copyright registration record for *ICICS-PCM 2003* that contains *Wang*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?Search_Arg=ICICS-PCM+2003+&Search_Code=TALL&PID=XPcJeBd2ye_USvtkZIc_2muZqjCmlh

Google Exhibit No. 1063
Page 0193

Exhibit	Description
	&SEQ=20250313101411&CNT=25&HIST=1 from the public catalog of the US Copyright Office.
1007E	Bibliographic and MARC records for *ICICS-PCM 2003* that contains *Wang*, available at https://catalog.lindahall.org/permalink/01LINDAHALL_INST/19lda7s/alma991911243405961 from the online catalog of the Linda Hall Library.
1007F	Profile of Linda Hall Library, available at https://librarytechnology.org/library/2294
1041	Wolters, M., Kjorling, K., Homm, D., & Purnhagen, H. (2003, October). A closer look into MPEG-4 High Efficiency AAC. In *Audio Engineering Society Convention 115*, Convention Paper 5871 ("*Wolters*").
1041A	Copy of *Wolters*, obtained from the Audio Engineering Society.
1041B	Bibliographic and MARC records for *Convention Papers : Presented at the 115th Convention, 2003 October 10-13, New York, New York* that contains *Wolters*, available at https://catalog.lindahall.org/permalink/01LINDAHALL_INST/19lda7s/alma993980363405961 from the Linda Hall Library.
1041C	Profile of Linda Hall Library, available at https://librarytechnology.org/library/2294
1041D	Bibliographic and MARC records for *Convention Papers : Presented at the 115th Convention, 2003 October 10-13, New York, New York* that contains *Wolters*, available at https://science-catalogue.canada.ca/record=2015693~S6 from the Canadian National Science Library.
1041E	AES 115th Convention Program Session C archived by Internet Archive on August 31 2003, available at https://web.archive.org/web/20030831165223/http://www.aes.org/events/115/papers/SessionC.cfm, and another copy archived by Internet Archive on Dec 17 2003, available at https://web.archive.org/web/20031217213041/http://www.aes.org/events/115/papers/SessionC.cfm.
1041F	"AES Special Publications" archived by Internet Archive on February 18, 2004, available at

Exhibit	Description
	https://web.archive.org/web/20040218195627/http://aes.org/publications/convention_cds.cfm
1041G	"Audio Engineering Society Preprints" archived by Internet Archive on June 15, 2004, available at https://web.archive.org/web/20040615041014/http://aes.org/publications/preprints/lists/115.cfm, and "Audio Engineering Society Preprints" order page archived by Internet Archive on April 4, 2004, available at https://web.archive.org/web/20040404224624/http://www.aes.org/publications/preprints/search.html
1041H	2004 citations to *Wolthers*, obtained from Google Scholar.
1005	Watkinson, J. (2001). *The MPEG Handbook : MPEG-1, MPEG-2, MPEG-4*. Focal Press. ("*Watkinson*")
1005A	Bibliographic and MARC records for *Watkinson* available at https://lccn.loc.gov/2001044789 from the online catalog of the Library of Congress.
1006	Gourley, D., & Totty, B. (2002). *HTTP : The Definitive Guide*. First Edition. O'Reilly. ("*Gourley*")
1006A	Bibliographic and MARC records for *Gourley*, available at https://lccn.loc.gov/2002023325 from the online catalog of the Library of Congress.
1006B	Copyright registration record for *Gourley*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?v1=1&ti=1,1&Search%5FArg=HTTP%20%3A%20The%20Definitive%20Guide&Search%5FCode=TALL&CNT=25&PID=Bdib-lihW_ZDN_qvinjMgCNi4cmCp6&SEQ=20250313101234&SID=7 from the public catalog of the US Copyright Office.
1014	A Guide to MPEG Fundamentals and Protocol Analysis (including DVB and ATSC) (2000). ("*MPEG Guide*")
1014A	*MPEG Guide* archived by Internet Archive on August 7, 2001, available at https://web.archive.org/web/20010807020249/http://www.tektronix.com/Measurement/cgi-bin/framed.pl?Document=/Measurement/App_Notes/mpegfund/&FrameSet=televi

Exhibit	Description
	sion; and *MPEG Guide* archived by Internet Archive on December 25, 2002, available at https://web.archive.org/web/20021225003145/http://www.tektronix.com/Measurement/cgi-bin/framed.pl?Document=/Measurement/App_Notes/mpegfund/&FrameSet=television
1018	Mietens, S. O., With, de, P. H. N., & Hentschel, C. (2004, March). New complexity scalable MPEG encoding techniques for mobile applications, *EURASIP Journal on Applied Signal Processing*, 2004(2), 236-252 ("*Mietens*").
1018A	Serial record for *EURASIP Journal on Advances in Signal Processing* that contains *Mietens*, available at https://wrlc-amu.primo.exlibrisgroup.com/permalink/01WRLC_AMU/1jvcmg9/alma99186427786404102 from the online catalog of the American University Library.
1018B	Springer Nature Link at https://link.springer.com/journal/13634/volumes-and-issues, the webpage for 2004(2) at https://link.springer.com/journal/13634/volumes-and-issues/2004-2, and the webpage for *Mietens* at https://link.springer.com/article/10.1155/S1110865704309091
1019	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 ("*Hui*").
1019A	Library copy of *Hui*, obtained from Library of Congress.
1019B	Bibliographic and MARC records for *IEEE Journal on Selected Areas in Communications*, available at https://lccn.loc.gov/83643067 from the online catalog of the Library of Congress.
1019C	Citations to *Hui*, obtained from Google Scholar.
1020	Symes, P. (2004). Digital Video Compression. McGraw-Hill. ("*Symes*")
1020A	Bibliographic and MARC records for *Symes*, available at https://lccn.loc.gov/2003065185 from the online catalog of the Library of Congress.

Exhibit	Description
1021	Pereira, F. C., & Ebrahimi, T. (2002). *The MPEG-4 Book*. Prentice Hall PTR. ("*Pereira*")
1021A	Bibliographic and MARC records for *Pereira*, available at https://lccn.loc.gov/2002070433 from the online catalog of the Library of Congress.
1021B	Copyright registration record for *Pereira*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?Search_Arg=MPEG-4+Book&Search_Code=TALL&PID=W3DyOQY7K2A5TLHBI1bJ3NCYFWgPTY&SEQ=20250313100426&CNT=25&HIST=1 from the public catalog of the US Copyright Office.
1022	Hunt, C., & Thompson, R. B. (1998). *Windows NT TCP/IP Network Administration*. First Edition. O'Reilly. ("*Hunt*")
1022A	Copyright registration record for *Hunt*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?v1=5&ti=1,5&Search%5FArg=Windows%20NT%20TCP%2FIP%20Network%20Administration&Search%5FCode=TALL&CNT=25&PID=eM7LOn3Xmg-sg3Yvfq6oBI5xOZleF2&SEQ=20250313100521&SID=3 from the public catalog of the US Copyright Office.
1022B	Bibliographic and MARC records for *Hunt*, available at https://lccn.loc.gov/2001274264 from the online catalog of the Library of Congress.
1023	Comer, D. E. (2000). *Internetworking with TCP/IP. Volume 1: Principles, Protocols, and Architecture*. Fourth edition. Prentice Hall. ("*Comer*")
1023A	Bibliographic and MARC records for *Comer*, available at https://lccn.loc.gov/00027439 from the online catalog of the Library of Congress.
1023B	Copyright registration record for *Comer*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?v1=13&ti=1,13&Search%5FArg=Internetworking%20with%20TCP%2FIP&Search%5FCode=TALL&CNT=25&PID=u8dh-oBwFZwTMus01mKAJvsIK4qajU&SEQ=20250313100645&SID=5 from the public catalog of the US Copyright Office.
1027	Stockhammer, T., Hannuksela, M. M., & Wiegand, T. (2003, July). H. 264/AVC in wireless environments, *IEEE Transactions on*

Exhibit	Description
	Circuits and Systems for Video Technology, *13*(7), 657-673 ("*Stockhammer*").
1027A	Library copy of *Stockhammer*, obtained from Library of Congress
1027B	Bibliographic and MARC records for *IEEE Transactions on Circuits and Systems for Video Technology* that contains *Stockhammer*, available at https://lccn.loc.gov/91642188 from the online catalog of the Library of Congress.
1028	Richardson, I. E. G. (2003). H.264 and MPEG-4 Video Compression. Wiley. ("*Richardson*")
1028A	Bibliographic and MARC records for *Richardson*, available at https://essex.primo.exlibrisgroup.com/permalink/44UOES_INST/ngp9lg/alma991004277709707346 from the online catalog of the University of Essex Library.
1028B	Bibliographic and MARC record for *Richardson*, available at https://catalog.library.tamu.edu/Record/in00001815737 from the online catalog of the Texas A&M University Libraries.
1029	Bourge, A., & Jung, J. (2004, January). Low-power H. 264 video decoder with graceful degradation. In *Visual Communications and Image Processing 2004*, Proceedings of SPIE Vol. 5308, pp. 372-383 ("*Bourge*").
1029A	Library copy of *Bourge*, obtained from Linda Hall Library.
1029B	Bibliographic and MARC records for *Visual Communications and Image Processing 2004* that contains *Bourge*, available at https://catalog.lindahall.org/permalink/01LINDAHALL_INST/19lda7s/alma993806573405961 from the online catalog of the Linda Hall Library.
1032	Cen, S., Walpole, J., & Pu, C. (1997, December). Flow and congestion control for internet media streaming applications. In *Multimedia Computing and Networking 1998*, Proceedings of SPIE Vol. 3310, pp. 250-264 ("*Cen*").
1032A	Copy of *Cen*, obtained from the Linda Hall Library.
1032B	Bibliographic and MARC records for *Multimedia Computing and Networking 1998* that contains *Cen*, available at https://catalog.lindahall.org/permalink/01LINDAHALL_INST/19lda7s/alma992821293405961 from the online catalog of the Linda Hall Library.

Exhibit	Description
1033	Lu, J. (2000, April). Signal processing for Internet video streaming: A review. *Image and Video Communications and Processing 2000,* Proceedings of SPIE Vol. *3974*, pp. 246-259 ("*Lu*").
1033A	Copy of *Lu*, obtained from the Linda Hall Library.
1033B	Bibliographic and MARC records for *Image and Video Communications and Processing 2000* that contains *Lu*, available at https://catalog.lindahall.org/permalink/01LINDAHALL_INST/19lda7s/alma99335 6813405961 from the online catalog of the Linda Hall Library.
1038	Wetzel, A. T., & Schell, M. R. (1996). Consumer applications of the IEEE 1394 Serial Bus, and A 1394/DV Video editing system. ("*Wetzel*")
1038A	Copy of *Wetzel*, obtained from the Linda Hall Library.
1038B	Front matter of *The Anatomy of a High-Performance Microprocessor : A Systems Perspective* whose companion CD-ROM contains *Wetzel*, published by the IEEE Computer Society, obtained from the Linda Hall Library.
1038C	Bibliographic and MARC records for *The Anatomy of a High-Performance Microprocessor : A Systems Perspective* whose companion CD-ROM contains *Wetzel*, available at https://catalog.lindahall.org/permalink/01LINDAHALL_INST/19lda7s/alma99298 4703405961 from the online catalog of the Linda Hall Library.
1038D	Bibliographic and MARC records for *The Anatomy of a High-Performance Microprocessor : A Systems Perspective* whose companion CD-ROM contains Wetzel, available at https://lccn.loc.gov/98017345 from the online catalog of the Library of Congress.
1038E	Copyright registration record for *The Anatomy of a High-Performance Microprocessor : A Systems Perspective* whose companion CD-ROM contains Wetzel, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?Search_Arg=Anatomy+of+a+High-Performance+Microprocessor&Search_Code=TALL&PID=svkzohrlPu0WzIt-vowLaEtyGJuPnA&SEQ=20250313101040&CNT=25&HIST=1 from the public catalog of the US Copyright Office.

Exhibit	Description
1039	Paik, W. K., & Hwang, S. Y. (1999, November). Design of a novel synthesis filter for real-time MPEG-2 audio decoder implementation on a DSP chip, *IEEE Transactions on Consumer Electronics*, *45*(4), 1119-1129 ("*Paik*").
1039A	Copy of *Paik*, obtained from the Library of Congress.
1039B	Bibliographic and MARC records for *IEEE Transactions on Consumer Electronics* that contains *Paik*, available at https://lccn.loc.gov/75643472 from the online catalog of the Library of Congress.
1039C	Copyright registration record for vol. 45 of *IEEE Transactions on Consumer Electronics*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?v1=24&ti=1,24&Search%5FArg=IEEE%20Transactions%20on%20Consumer%20Electronics&Search%5FCode=TALL&CNT=25&PID=F6rD-dgau3TxaR9C7h4R2WLEmmSh_I&SEQ=20250313102127&SID=10 from the public catalog of the US Copyright Office.
1040	Lee, K. S., Park, Y. C., & Youn, D. H. (2002, August). Software optimization of the MPEG-audio decoder using a 32-bit MCU RISC processor, *IEEE Transactions on Consumer Electronics*, *48*(3), 671-676 ("*Lee*").
1040A	Copy of *Lee*, obtained from the Library of Congress.
1040B	Bibliographic and MARC records for *IEEE Transactions on Consumer Electronics* that contains *Lee*, available at https://lccn.loc.gov/75643472 from the online catalog of the Library of Congress.
1042	De Smet, P., Rooms, F., Luong, H. Q., & Philips, W. (2004, October). Do not zero-pute: An efficient homespun MPEG-Audio layer II decoding and optimization strategy. In *2004 multimedia conference : MM'04, Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 376-379 ("De Smet 2004").
1042A	FirstSearch record for *2004 multimedia conference : MM'04* that contains *De Smet 2004*, available to subscribers on OCLC FirstSearch at https://firstsearch.oclc.org/.
1043	De Smet, P, & Bruyland, I. (2003, October). Optimized recursive subband synthesis windowing for implementing efficient MPEG audio decoders, *IEEE Signal Processing Letters*, vol. 10, no. 10, 303-306 ("*De Smet 2003*").
1043A	Copy of *De Smet 2003*, obtained from the Library of Congress.

Exhibit	Description
1043B	Bibliographic and MARC records for *IEEE Signal Processing Letters* that contains *De Smet 2003*, available at https://lccn.loc.gov/94648235 from the online catalog of the Library of Congress.
1043C	Copyright registration record for volume 10 of *IEEE Signal Processing Letters* that contains *De Smet 2003*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?v1=7&ti=1,7&Search%5FArg=IEEE%20Signal%20Processing%20Letters&Search%5FCode=TALL&CNT=25&PID=Uo0kv8IbGlbO7UAdxPRsL-ABRfe3Du&SEQ=20250313102254&SID=11 from the public catalog of the US Copyright Office.
1044	Overview of the MPEG-4 standard. Robert Koenen (editor). ISO/IEC JTC1/SC29/WG11 N4668. March 2002. Version 21 ("*MPEG Overview*")
1044A	*MPEG Overview* archived by Internet Archive on June 4, 2002, available at https://web.archive.org/web/20020604024616/http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm.
1047	Mesarovic, V., Rao, R., Dokic, M., & Deo, S. (2001, April). Optimized Huffman Decoding for AAC. In *Audio Engineering Society Conference: UK 16th Conference: Silicon for Audio*. Audio Engineering Society, pp. 117-122 ("*Mesarovic*").
1047A	FirstSearch record for *Silicon for Audio : Audio Engineering Society 16th UK conference* that contains *Mesarovic*, available to subscribers on OCLC FirstSearch at https://firstsearch.oclc.org/.
App A	Hsieh-Yee CV
App B	Parts VII to X of Understanding MARC Bibliographic, available at https://www.loc.gov/marc/umb/um07to10.html from the Library of Congress.

COMPUTATIONAL GRACEFUL DEGRADATION FOR VIDEO DECODING

Marco Mattavelli, Sylvain Brunetton
Integrated Systems Center - C3i,
Swiss Federal Institute of Technology,
Lausanne CH-1015, SWITZERLAND
Tel: +41 21 6936979; fax: +41 21 6934663
e-mail: Marco.Mattavelli@epfl.ch

ABSTRACT

The implementation of software based video decoders is an advantageous solution over traditional dedicated hardware real-time systems. However, guaranteeing real-time performance, needed when processing video/audio bit-streams, by scheduling processing tasks of variable load and that might exceed the available resources, is an extremely difficult and in many cases an impossible challenge. In this paper we introduce the motivations and ideas of the Computational Graceful Degradation (CGD) approach for the implementation of video decoding. The goal is to provide results of possibly lower visual quality, but respecting the real-time constraints, when the required processing needs exceed the available processing resources. This approach is very attractive since it enables a more efficient usage of the hardware processing power without needing to implement resources for worst case complexity decoding. We show how such technique can be applied to the decoding of compressed video sequences. Simulation results for MPEG-4 and H.263 video compression standards showing some of the interesting achievements and potentialities of CGD approach are also reported. The presented approach can be virtually applied to any video compression standard and processor based platform, although the best performances are achieved when complexity prediction information provided as side information by the video standard is available. This option has been included in the new MPEG-4 standard.

1 INTRODUCTION

The complexity of video decoding is a fluctuating function of time [1]. By complexity we mean the processing time necessary to decode a compressed video bitstream. Even considering a specific processor platform, a specific compression standard, with a relative level and profile, the complexity of video decoding is a variable quantity. Dedicated hardware platforms can guarantee worst case complexity decoding. In contrast, in the case of programmable platforms such a guarantee would imply the allocation of a processing power up to two orders of magnitude higher than the one actually needed for an average decoding [1]. Conversely, a theoretical average value is not useful for an efficient scheduling of a decoding task. The average value for each considered sequence is not stationary, and its variation around the theoretical average one are so unpredictable that no useful scheduling policy can be realized. The first straightforward option is a sustained over-sizing of the processing resources. The second is to accept interruptions in the audio/video output streams when the needed decoding resources exceed the ones available in the system. Obviously both options are not desirable being respectively not economical, or providing a quality of the services which may easily result perceptually unacceptable. The main rationale of the Computational Graceful Degradation (CGD) approach is to provide results of poorer quality when the results of the desirable quality cannot be obtained in real time. When the complexity of the task exceeds the available processing power, real-time services, possibly of degraded quality, are nevertheless guaranteed respecting all real-time constraints. This approach enables the correct scheduling of real-time tasks even when the computational system load fluctuates above the system capabilities.

The concept of computational graceful system degradation using imprecise results has previously been introduced in the field of real-time computing with applications to control system and automation. In these contexts imprecise results are defined as intermediate results with the property that the more recent and intermediate result is the closer one that approximates the final result (i.e. the idea only works in monotonically improving computations) [2, 3, 4]. In these cases the problems to be solved are more concerned with fault tolerance, I/O overloads, stability and reactivity to unexpected inputs, and the corresponding techniques are not suitable for the purposes of video decoding. In our case, the goal is to guarantee real-time video decoding with possible degradations of the image quality, i.e reductions of the peak signal to noise ratio (PSNR), without loosing parts of the image content. Such al-

Google Exhibit No. 1004
Page 001

Google Exhibit No. 1063
Page 0202

ternative is certainly much more desirable than other straightforward options such as cutting image portions or simply skipping frames. These latter options, as can be easily understood, can have catastrophic consequences on prediction based coding systems. In H.263 decoders, for instance, skipping the decoding of a single frame or a portion of it, implies the skipping of all future incoming frames (or portion of them) before the next incoming Intra frame. The time interval of such skipping operation might cover many seconds and the information content loss might be very relevant. Such abrupt interruptions, which are perceptually very annoying, might be caused not only by the simple fluctuation of the decoding complexity load, but also by the fluctuations of the load of other non-video processing tasks which reduce the available processing resources. Moreover, such events in general are out of the control of the video decoder system and cannot be easily forecasted. Considering these facts, it is evident the importance of techniques able to filter out complexity decoding peak loads or able to reduce the average decoding computation load without causing perceptually annoying consequences. Skipping B-VOP could be, in some cases, a viable option accepting the introduction of motion judder in the sequence reproduction. Unfortunately the presence of B-frames or B-VOP cannot be guaranteed in any bit-stream and just at the right time for which a reduction of the complexity is needed to respect the real time constraints.

In contrast with what could be commonly thought, also the classical multi-resolution coding features such as spatial or temporal scalabilities are not really useful for our goals. In general, they are used to optimize the transmission bandwidth usage enabling the compatibility of services at different titrates and spatial resolutions, but they must be embedded in the original bit-stream at the encoder side. If they are not embedded, the changes of resolution at the decoder size can results more computationally expansive than the simple decoding at the given resolution.

From these arguments the usefulness of CGD and its important features and applications are clear. The last key point that has left CGD as a good idea, but of difficult introduction in video coding, was that it was not clear how to implement it efficiently. Now that a technique able to predict with a good degree of approximation the required decoding time of each decoding process is demonstrated and available for MPEG-4 systems [5], we have the key for the useful and efficient implementation of CGD techniques.

2 PRINCIPLES OF DECODING USING THE CGD APPROACH

The decoding operations of a video compression algorithm can be divided into two main categories: 1) operations that cannot be simplified since they may prevent the extraction and usage of relevant coding information, 2) operations that can be simplified, possibly leading to imprecise or degraded video results. To the first category typically belong all bit-stream parsing operations, entropy decoding (i.e. VLD and arithmetic coding), and all operations related to synchronization and to the syntax of the decoding process. For these operations no degradation mode definition is possible, otherwise it would imply skipping image portions or entire frames since either the data or the coding modes are lost. To the second category belong operations such as the IDCT, the prediction with and without motion compensation in its various forms (PB/B-frame prediction, overlapped block motion compensation (OBMC), half pixel interpolation, macroblock padding and so on). Simplifications of such operations applied to pixels or transform coefficients are always possible but at the expenses of imprecise results.

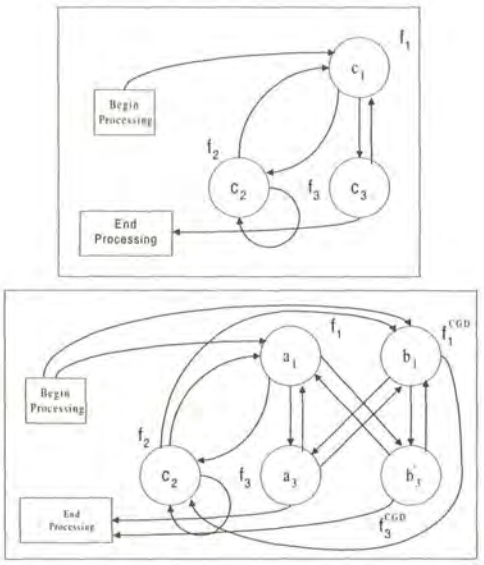


Figure 1: Upper diagram, example of a complexity model representing the data dependent execution of three algorithms. Lower diagram, example of the same complexity model of the upper diagram for which two algorithms (f_1 and f_3 are associated to new CGD algorithms (f_1^{CGD} and f_1^{CGD}).

Two are the possible strategies for the implementation of CGD features. The first is to monitor the execution of the decoding process, to assume that the decoding time of each portion of the frame or VOP is proportional to the image area, and to execute CGD algorithms whenever the current monitored decoding time exceeds the current target decoding time suitable to respect the specified real-time constraints. Such simple strategy has several drawbacks. One is that there is no guaran-

Google Exhibit No. 1004
Page 002

Google Exhibit No. 1063
Page 0203

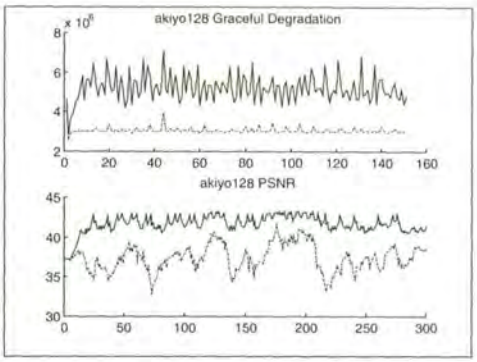


Figure 2: Example of CGD decoding of 300 frames of the sequence Akiyo at compressed at 128 kb/sec with H.263. The upper diagram reports the number of processor cycles for the original decoding (upper curve) and for the CGD decoding lower curve, versus the frame number. The lower diagram reports the two PSNR.

tee that the target decoding time is achieved. Another problem is due to execution of CGD algorithms that *a posteriori* could have not been necessary, thus introducing unnecessary image degradations. Moreover, such degradations are concentrated on critical image areas, and due to the drift given by the degraded reference for the prediction based algorithms, such degradations becomes visible and might seriously affect the resulting sequence PSNR. An example of such approach is reported in Fig. 2. The H.263 decoder has been modified by implementing computationally degraded IDCT and motion compensation algorithms. The interesting result achieved is that the decoding is obtained at a constant processing time for each frame. Unfortunately the PSNR degradations although limited to some dBs are nevertheless relevant from the perceptual point of view.

The second strategy is to exploit the complexity information carried by the frames or VOP headers, like it can be done for MPEG-4 systems [6], and use it in two stages. The first stage predicts the decoding time needed for the current frame or VOP, as described in [5]. The second, on the basis of the predicted decoding time and on the complexity information (i.e. statistics of the algorithms necessary to correctly decode the frame), executes only the CGD algorithms that are just necessary to not exceed the target decoding time.

More formally, let us consider a complexity model similar to the one illustrated in Fig. 1 composed by a set of I algorithms f_i, with $i = 1 \dots I$, and suppose to be able to design for all, or only for a subset S^{CGD} of the functions f_i composing the model, new functions f_i^{CGD} which have the following three properties: 1. the execution of a function f_i^{CGD} yields results that are an approximation of the ones produced by the original function f_i, 2. the execution of f_i^{CGD} does not prevent the execution of any subsequent decoding functions, 3. the weight w_i^{CGD} associated to f_i^{CGD} that represent the execution time of the f_i^{CGD} algorithm for the considered decoder implementation is lower than the corresponding w_i associated to f_i (i.e. $w_i^{CGD} < w_i$). A schematic representation of a complexity model which includes new degraded functions is reported in Fig. 1. The picture shows that new degraded algorithms can be executed at the place of the original algorithms. Obviously, the execution of any f_i^{CGD} may produce PSNR degradations in the reconstruction of some image parts, it does not prevent from correctly decoding the rest of the frame or VOP, but it reduces the necessary decoding time. Let us partition the functions constituting the complexity model into two disjoint sets: S^{Orig} and S^{CGD} respectively for the ones that do not admit a CGD version (f_2 for instance in Fig. 1) and for the ones for which it is possible to design a CGD version (f_1 and f_3). Let us consider for simplicity an image sequence composed by a sequence of single frames or VOPs and suppose that the execution time for the decoding of the frame or VOP is larger than the allocated decoding time:

$$T(t) > T_{Alloc}, \tag{1}$$

where T(t) is the decoding time of the VOP at hand at the sampling time t. Since the decoding time can be approximated by its prediction [5] it can be expressed by:

$$T(t) \simeq T_{Pred}(t) = \sum_{i=1}^{I} c_i w_i, \tag{2}$$

where the c_i are the actual occurrences of algorithm f_i in the current frame or VOP [5], and the w_i are the corresponding complexity weights. The lower limit for the decoding time executing all CGD algorithms f_i^{CGD} at the place of f_i for all $i \in S^{CGD}$, is given by:

$$T_{CGD} = \sum_{i \in S_{CGD}} c_i w_i^{CGD} + \sum_{i \in S_{Orig}} c_i w_i. \tag{3}$$

If $T_{CGD} < T_{Alloc} < T_{Pred}$, is possible to guarantee the real-time constraints by executing the f_i^{CGD} algorithms and accepting approximate decoding results. It is also possible to find for each c_i with $i \in S^{CGD}$, a couple of coefficients a_i, b_i, for which $a_i + b_i = c_i$, and denoting as $T_{Process}$ any decoding time interval which respects the real time constraints, we can express the desired process decoding time as any decoding time value that satisfy the expression:

$$T_{process} = \sum_{i \in S_{CGD}} (a_i w_i^{CGD} + b_i w_i) + \sum_{i \in S_{Orig}} c_i w_i \leq T_{Alloc}. \tag{4}$$

Obviously for any Allocated Decoding Time the goal is to respect the real-time constraints with the lowest

Google Exhibit No. 1004
Page 003

Google Exhibit No. 1063
Page 0204

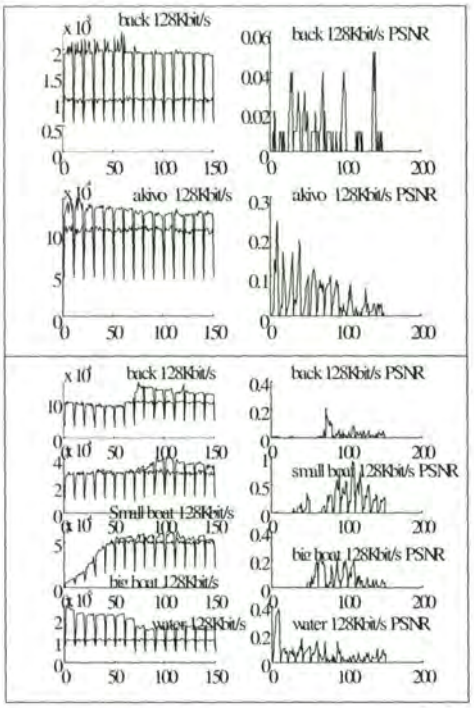


Figure 3: Left, decoding time results for the various VOP of sequences Akiyo and Coastguard, with and without imposing a bound. Right, the PSNR degradations versus the sequence VOP number.

degradation in the video/audio content. Associating a degradation measure to each w_i^{CGD} it is possible to minimize the degradation for any given T_{Alloc}. The challenge is to find the set $S_{a,b}$ of couple of coefficients a_i, b_i, with $i \in S_{CGD}$, that satisfy minimize the video degradations expressed as difference with the non degraded output:

$$S_{a,b} = \text{argmin(a, b)}_{i \in S_{CGD}} : (\text{PSNR} - \text{PSNR(a, b)}). \quad (5)$$

The determination of the set $S_{a,b}$ can be done by means of many suitable optimization method or heuristics. The described techniques can be easily extended to the case of sequences containing multiple VOP at the same sampling time t.

3 SIMULATION RESULTS

Simulation results demonstrate the possibility of successful implementations of CGD for MPEG-4 video decoding. A bound on the decoding complexity of each VOP is defined setting a maximum constant T_{Alloc}. Figure 3 report the results of MPEG-4 video decoding imposing such bounds on the decoding time of each video object plane (VOP) for the test images Coastguard and Akiyo coded at 128 kb/s. On the left, the graphs report the decoding time in msec for the unbounded and bounded decoding (lower curve) versus the image sequence number. On the right the graphs reports the resulting PSNR degradation. The decoder is the Momusys CD on a Sun Ultrasparc station modified implementing degraded motion compensation algorithms. Obviously the actual results depend on the decoder, on the platform on the CGD algorithms implemented, but the good results demonstrate that bounded complexity decodings are possible with minimal image degradations much lower than the one obtained without complexity information and decoding time prediction.

4 CONCLUSION

We have introduced the motivations and concepts of the CGD approach for the implementation of video decoders on processor based platforms. Simulation results for MPEG-4 video decoding demonstrate the successful implementation of the CGD approach. Obviously, the implementation of CGD features are not proposed to be used at their maximum degradation rate but as tool able to guarantee real-time performance in presence of peaks of decoding complexity or as a way to ease the scheduling problems given by temporary fluctuations of the available processing power. Moreover, the presented approach may enable the successful implementation of video decoding for levels and profiles that would be forbidden if the complexity of *worst case decoding* has to be guaranteed.

References

[1] M. Mattavelli and S. Brunetton. Scalability of the mpeg-4 vm based on the computational power of texture decoding. Technical Report Document M925, ISO/IEC JTC1/SC29/WG11, Tampere, Finland, July 1996.

[2] A.D. Stoyenko W.A. Halang. *Constructing predictable real time systems.* Kluwer Academic Publisher, 1991.

[3] J. Liu K. Lin, S. Natarjan. Imprecise results: Utilizing partial computations in real-time systems. In *Proc. of IEEE Real-Time System Symp.*, Dec. 1987.

[4] S. Natarjan J. Liu, K. Lin. Scheduling real time periodic jobs using imprecise results. In *Proc. of IEEE Real-Time System Symp.*, Dec. 1987.

[5] M. Mattavelli and S. Brunetton. Prediction of decoding time for mpeg-4 video. In *Proc. of Eusipco 98, IX European Signal Processing Conference*, Island of Rhodes, Greece, September 8-11 1998.

[6] ISO-IEC/JTC1. Doc for iso/iec cd 14496-2 visual. Technical Report Doc. W2119, Tokyo, March 1998.

Google Exhibit No. 1004
Page 004

Google Exhibit No. 1063
Page 0205

EUROPEAN
SIGNAL
PROCESSING
CONFERENCE

SIGNAL
PROCESSING
IX

TK 5102
.9
.E97
1998
Vol 4

Google Exhibit No. 1004
Page 005

Google Exhibit No. 1063
Page 0206

SIGNAL PROCESSING IX

Theories and Applications

VOLUME IV



Edited by:

S. Theodoridis, I. Pitas,

A. Stouraitis, N. Kalouptsidis

TYPORAMA PUBLICATIONS

Google Exhibit No. 1004
Page 006

Google Exhibit No. 1063
Page 0207

SIGNAL PROCESSING IX
THEORIES AND APPLICATIONS

Google Exhibit No. 1004
Page 007

Google Exhibit No. 1063
Page 0208

SIGNAL PROCESSING IX

THEORIES AND APPLICATIONS

Proceedings of Eusipco–98

Ninth European Signal Processing Conference

Rhodes, Greece, 8–11 September 1998

VOLUME IV OF IV

Edited by

S. Theodoridis, I. Pitas, A. Stouraitis, N. Kalouptsidis



UNIVERSITY OF ATHENS



COMPUTER TECHNOLOGY INSTITUTE, PATRAS



THE EUROPEAN ASSOCIATION FOR SIGNAL PROCESSING

TYPORAMA EDITIONS, 1998

Eusipco–98 Proceedings in 4 Volumes
Printed in Greece
ISBN: 960–7620–05–4
960–7620–06–2 (Vol. I)
960–7620–07–0 (Vol. II)
960–7620–08–9 (Vol. III)
960–7620–09–7 (Vol. IV)

© Eurasip
Typorama Editions

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher.
No responsibility is assumed by the publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

Typorama Editions: Iroon Polytechniou 110, GR–264 42 Patras, Greece — Ph: +3061 430730, Fax: +3061 430884 — info@typorama.gr

Google Exhibit No. 1004
Page 008

Google Exhibit No. 1063
Page 0209

LIBRARY OF CONGRESS
DEC 16 1998
GIFT

TK 5102
.9
.E97
1998
vol.4

98-232284

General Chairman's Message

It is my pleasure to warmly welcome you all to EUSIPCO-98 in the beautiful Greek island of Rhodes. A place rich in history, culture, and natural beauties.

EUSIPCO-98 is to be held at the RODOS PALACE International Convention Center. It is located by the sea and it is an ideal place to combine stimulating scientific meetings with leisure.

I am very pleased that the response of the Signal and Image Processing Scientific Community confirmed the tradition of continued growth of the EUSIPCO conferences. This year we received more than 960 submissions from all over the world, 86 of which were allocated in 12 special sessions, 270 in 38 oral sessions, and 274 to 21 poster sessions. The program of the conference also includes 4 plenary talks on "hot" topics in the areas of Signal and Image Processing as well as in Communications.

I would like to express my deep appreciation to all organizing committee members and in particular N. Kalouptsidis, I. Pitas and A. Stouraitis, for their efforts and invaluable help in organizing EUSIPCO-98. I would like to thank the International Scientific Committee members for offering their time in reviewing the submitted papers. Their effort made it possible for the papers to be reviewed by three reviewers independently, which guaranteed a rigorous reviewing process, matching the prestige of the EUSIPCO conferences. Last but not least I want to thank the plenary speakers, the special session organizers, the session chairmen, and all the authors for their efforts towards a successful scientific event.

On behalf of the organizing committee, I wish you a pleasant stay in Rhodes, and I hope that all of you will enjoy an experience rich in science, culture and natural beauty.

Sergios Theodoridis
General Chairman

Google Exhibit No. 1004
Page 009

Google Exhibit No. 1063
Page 0210

Organizing Committee

General Chairman
S. THEODORIDIS
University of Athens
stheodor@di.uoa.gr

Co-chairman
N. KALOUPTSIDIS
University of Athens
kalou@di.uoa.gr

Program Chairman
I. PITAS
University of Thessaloniki
pitas@zeus.csd.auth.gr

Co-chairman
A. STOURAITIS
University of Patras
thanos@ee.upatras.gr

Special Sessions
K. BERBERIDIS
C.T.I.
berberid@cti.gr

Tutorials
G. MOUSTAKIDES
University of Patras
moustaki@cti.gr

Invited Speakers
G. CARAYANNIS
I.L.S.P.

Publications
S. KOLLIAS
National Technical University of Athens

Publicity
A. SKODRAS
University of Patras

Exhibitions
G. KOUROUPETROGLOU
G. GLENTIS
University of Athens

Finance
R. EFSTATHIADOU
L. GOURDOUPI
C.T.I.

Local Arrangements
S. KARABOYAS
E. SAGRIOTIS
University of Athens
K. TSOLAKIDES
Univ. of Aegean

International Liaisons
J. PROAKIS (USA)
A. VENETSANOPOULOS (CD)
H. SAKAI (J)

Google Exhibit No. 1004
Page 010

Google Exhibit No. 1063
Page 0211

Organization

Organizer:

EUROPEAN ASSOCIATION FOR SIGNAL PROCESSING

Sponsors:

NOKIA HELLAS S.A.

ALPHACOPY S.A.

ERICSSON HELLAS S.A.

SIEMENS S.A.

INTRAKOM S.A.

THE HELLENIC TELECOMMUNICATIONS ORGANIZATION (OTE)

THE MINISTRY OF EDUCATION

THE GENERAL SECRETARIAT FOR SCIENCE AND TECHNOLOGY

THE MINISTRY OF THE AEGEAN

THE MINISTRY OF CULTURE

THE CITY OF RODOS

Best paper Awards Sponsors:

ALCATEL ITALIA S.A.

ELSEVIER SCIENCE B.V.

Exhibitor:

MENTOR HELLAS LTD.

International Scientific Committee

ACCARDO A.
ADALI T.
AHALT S.
ALVERTOS N.
ARCE G.
ASTOLA J.
BANGHAM J.A.
BARLAUD M.
BATALAMA S.
BAYOUMI M.A.
BELLANGER M.
BENVENUTO N.
BERBERIDIS K.
BERSHAD N.
BIEMOND J.
BIGLIERI E.
BISIANI R.
BISIO G.M.
BLOCH I.
BOHME J.F.
BOLON P.
BORGEFORS G.
BOROCZKY L.
BORS A.
BOUMAN C.A.
BOVIK A.
BROFFERIO S.C.
BURKHARDT H.
BUSCH
CAELEN J.
CAFFORIO C.
CAIN G.
CAPPELLINI V.
CASTANIE F.
CETIN E.
CHAMZAS C.
CHASSERY J.M.
CLARKE R.J.
COMON P.
CONSTANTINIDES A.
COWAN C.
DAVIES R.
DE MOOR B.
DE MORI R.
DEL RE E.
DELIGIANNIS T.
DELLEPIANE S.
DELOPOULOS A.
DENG D.
DEPRETTERE E.
DEWILDE P.
DI CLAUDIO E.
DIAMANTARAS K.I.
DIGALAKIS V.
DOCAMPO D.
DUHAMEL P.
DURRANI T.
EBRAHIMI T.
EFSTRATIADES S.
EKLUNDH J-O.
EMILIANI P-L.
FAVIER G.
FAZEKAS K.
FIGUEIRAS-VIDAL A.R.
FLANDRIN 0.
FOTOPOULOS S.
GABBOUJ M.
GALATSANOS N.P.
GERSHO A.
GEZERLIS V.
GHARAVI H.
GIANNAKIS G.
GILLOIRE A.
GIUSTO D.
GLENTIS G.
GOUTSIAS J.
GRANLUND G.H.
GRANT P.
GUERIN-DUGUE A.
HASLER M.
HERAULT J.
HERO A.O.
HESS W.
HEUTE U.
HLAVAC V.
HOLT MURRAY J.J.
JONES S.
JULLIEN G.A.
JUTTEN C.
KABOURAS D.
KAISER J.F.
KALOUPTSIDIS N.
KATSAGGELOS A.
KAUP A.
KIDA T.
KITTLER J.
KOLLIAS S.
KOTROPOULOS C.
KOUROUPETROGLOU G.
KOUTROUBAS K.
KROPATSCH W.G.
KUBIN G.
KUNT M.
LAFACE P.
LAGENDIJK R.L.
LANCINI R.
LE ROUX J.
LEONARDI R.
LIAVAS A.
LIND L.
LUETTIN J.
MACCHI O.
MACQ BENOIT M.
MAITRE H.
MALOBERTI F.
MARAGOS P.
MARSHALL S.

Google Exhibit No. 1004
Page 011

Google Exhibit No. 1063
Page 0212

Martinelli G.
Mathews V.J.
McCanny J.
McLaughlin S.
McWhirter J.G.
Mecklenbraeuker W.
Mersereau R.M.
Mertzios B.
Meyer F.
Mian G.A.
Mitra S.
Moonen M.
Moore R.
Mouriopoulos
Moustakides G.
Mulgrew B.
Mumolo E.
Najim M.
Nandi A.
Nickel U.
Noll P.
Onural
Orchard M.
Papamarkos
Papamichalis P.
Pappas T.
Parhi K.K.
Pattichis K.
Pereira F.M.
Perkis A.
Petrou M.
Picinbono B.
Pitas I.
Pols L.
Proakis J.G.
Proudler I.K.
Ramponi G.
Ramstad T.
Regalia P.
Regazzoni C.S.
Roberto V.
Rocca F.
Roerdink J.B.T.M.
Safranek R.J.
Sagriotis M.
Sakai H.
Salembier P.
Sandham W.
Sankur B.
Sayed A.
Scarano G.o
Serpico S.B.
Serra J.C.
Sezan M.I.
Sicuranza G.
Skodras A.N.
Skordalakis
Soderstrom T.
Sommer G.
Stoica P.
Stouraitis A.
Strintzis M.
Taguchi A.
Tasic J.F.
Tekalp A.M.
Tewfik A.H.
Theodoridis S.
Torres L.
Trahanias P.
Trancoso I.
Tsatsanis M.
Tziritas G.
Unbehauen H.
Unser M.
Vandewalle J.
Varkonyi-Koczy A.R.
Venetsanopoulos A.
Vernazza G.
Vesin J-M.
Wang P.
Woods J.W.
Yegnanarayana B.
Zamperoni P.
Zervakis M.E.
Zervas V.

Contents

Volume I

Table of contents ..... XIX–LXII

Proceedings 1–526

Watermarking, Copyright Protection and Access Control for Multimedia Systems 9–28
Combined Source and Channel Coding 29–40
Image and Video Coding 41–60
Spectral Estimation 61–80
Speech Enhancement 81–88
Audio and Electroacoustics I 89–132
Transforms and Signal Compression 133–176
Array Signal Processing 177–220
Video Compression and Communications 221–260
Multirate Systems 261–280
Filter Design 281–298
Face Recognition 299–318
Image Restoration 319–338
Speech Recognition 339–378
Pattern Recognition and Neural Networks 379–458
Architectures I 459–526

Author index ..... i–viii

Volume II

Table of contents ..... XIX–LXII

Proceedings 527–1212

Nonlinear Signal Processing for Communications 537–576
Shape Analysis and Recognition 577–616
Time – Frequency Analysis 617–656
Biomedical Signal Processing 657–676
Biomedical Image Processing 677–696
Speech Analysis 697–716
Speech and Audio Coding / Compression 717–740

Google Exhibit No. 1004
Page 012

Google Exhibit No. 1063
Page 0213

Efficient Algorithms for Filter Design and Signal Reconstruction *741–792*
Image Filtering *793–828*
Image Restoration and Enhancement *829–864*
Architectures II *868–904*
Motion *905–944*
Acoustic Echo and Noise Control *945–984*
Statistical Signal Processing *985–1004*
Parameter Estimation *1005–1028*
Image Enhancement *1029–1048*
Image Quality Evaluation *1049–1072*
3–D Imaging *1073–1112*
Applications and Architectures *1113–1132*
Speech Processing *1133–1212*

AUTHOR INDEX ..... i–viii

Volume III

TABLE OF CONTENTS ..... XIX–LXII

PROCEEDINGS 1213–1936

Adaptive Systems I *1221–1260*
3–D Imaging and Virtual Inspection of Objects and Places *1261–1296*
Neural Networks and Fuzzy Systems *1297–1332*
Signal Processing for Communications I *1333–1372*
Nonlinear Signal Processing and Systems *1373–1412*
Speech Coding *1413–1452*
Speech Enhancement / Recognition *1453–1492*
Image and Video Coding *1493–1544*
Motion Estimation *1545–1580*
Pattern Recognition *1581–1600*
Nonlinear Techniques for Channel Equalization *1601–1624*
Blind Identification / Deconvolution *1625–1664*
Image Analysis *1665–1704*
Multimedia Signal Processing *1705–1736*
Transform Methods in Signal Processing *1737–1772*
Statistical and Array Signal Processing *1773–1820*
Spectral Estimation and Signal Analysis *1821–1868*
Filter Design and Subband Techniques *1869–1936*

AUTHOR INDEX ..... i–viii

Volume IV

TABLE OF CONTENTS ..... XIX–LXII

PROCEEDINGS 1937–2568

Applications of Multirate Digital Signal Processing *1943–1976*
Signal Processing for Communications II *1977–2008*
Simulation Based Computational Methods for Statistical Signal Processing *2009–2036*
Adaptive Systems II *2037–2056*
Blind Source Separation *2057–2096*
Motion / Compression *2097–2112*
System Identification and Parameter Estimation *2113–2164*
Image and Shape Analysis *2165–2216*
Signal Processing for Communications III *2217–2252*
Image Filtering *2253–2272*
Watermarking *2273–2292*
Image and Video Coding *2293–2332*
Audio and Electroacoustics II *2333–2368*
System Identification *2369–2388*
Higher Order Statistics *2389–2412*
Applications *2413–2492*
Image Analysis *2493–2568*

AUTHOR INDEX ..... i–viii

Google Exhibit No. 1004
Page 013

Google Exhibit No. 1063
Page 0214

Table of Contents

Nomenclature

WA SS–3 **T** : Tuesday
W : Wednesday
Th : Thursday
F : Friday

WA **S**S–3 **M** : Morning
A : Afternoon

WA **SS**–3 **S** : Session
SS : Special Session
P : Poster Session

WA SS–**3** **1–7**: Rank of Presentation

Volume I

PLENARY TALK

Biometric Image Processing and Recognition 1
J. Phillips, R. Michael McCabe and Rama Chellappa

TM SS–1

Watermarking, Copyright Protection and Access Control for Multimedia Systems

Digital Watermarking: An Overview 9
G. Voyatzis, N. Nikolaidis and I. Pitas

On the Reliability of Detecting Electronic Watermarks in Digital Images 13
Ton Kalker, Jean–Paul Linnartz, Geert Depovere and Maurice Maes

A M.A.P. Identification Criterion for DCT–Based Watermarking 17
M. Barni, F. Bartolini, V. Cappellini, A. Piva and F. Rigacci

A Blind Wavelet Based Digital Signature for Image Authentication 21
Liehua Xie and Gonzalo R. Arce

Robust audio watermarking in the time domain 25
P. Bassia and I. Pitas

TM SS–2

Combined Source and Channel Coding

Optimal and Sub–Optimal Decoding for Vector Quantization over Noisy Channels with Memory 29
Mikael Skoglund

Approximating the protection offered by a channel code in terms of bit error rate 33
Fabrice Labeau, Claude Desset, Benoît Macq and Luc Vanderdorpe

Joint Binary Symmetric Source–Channel Coding with Small Linear Codes 37
François–Xavier Bergot and Olivier Rioul

TM S–3

Image and Video Coding

Image Compression Utilizing the DTOCS Topological Raster Patterns and Delaunay Triangulation 41
Pekka J. Toivanen

Google Exhibit No. 1004
Page 015

Google Exhibit No. 1063
Page 0216

Polyphase Adaptive Filter Banks for fingerprint image compression 45
Ömer N. Gerek and A. Enis Çetin

Universal context modeling for lossy wavelet image compression 49
Bernd Menser and Frank Müller

Hybrid coding of video with spatio–temporal scalability using subband decomposition 53
Marek Domański, Adam Łuczak, Sławomir Maćkowiak and Roger Świerczyński

Fast embedded video compression using cache–based zerotree processing 57
Charles D. Creusere

TM S–4
Spectral Estimation

Facts and Fiction in spectral Analysis of Stationary Stochastic Processes 61
P.M.T. Broersen

A Fast Algorithm for Efficient Estimation of Frequencies 65
Ta–Hsin Li

The generalized ACM–MUSIC without estimation of the number of sinusoidal components 69
Olivier Caspary and Patrice Nus

On nonparametric Spectral Estimation 73
Tomas Sundin and Petre Stoica

New criteria bases on Gerschgorin radii for source number estimation 77
Olivier Caspary and Patrice Nus

TM S–5
Speech Enhancement

Speech Enhancement in Wireless Digital Communication via Heuristic Rules and Image Relaxation Techniques 81
Enzo Mumolo, Lorenzo Pivetta and Claudio Chiaruttini

Analytical and Iterative Approaches to the Equalisation of Sub–Band Errors in Speech and Speaker Recognition 85
Ronald Auckenthaler and John S. Mason

On real time implementation aspects of a source separation algorithm 1465
Ulf A. Lindgren

TM P–6
Audio and Electroacoustics I

Multirate Acoustic Echo Cancellation: Which Adaptive Filters for Which Subbands? 89
Desmond K. Phillips and Colin F.N. Cowan

Robust Performance of the Adaptive Periodic Noise Canceller in a Closed–Loop System 93
J. Timoney and J.B. Foley

A Binaural System for the Suppression of Late Reverberation 97
K. Lebart, J.M. Boucher and P.N. Denbigh

The All–Pass Filtered–X Algorithm 101
John Garas and Piet C.W. Sommen

Multichannel Noise Reduction —Algorithms and Theoretical limits— 105
Joerg Bitzer, Klaus Uwe Simmer and Karl–Dirk Kammeyer

Design of a Transform Coder For High Quality Audio Signals 109
Mohammed Javed Absar, George Sapna and Antonio Alvarez

A New Subband Perceptual Audio Coder using CELP 113
Olivier van der Vrecken, Laurent Hubaut and Florence Coulon

Auditory Modeling via Frequency Warped Wavelet Transform 117
Gianpaolo Evangelista and Sergio Cavaliere

Wavelet Transform Based Coherence Function for Multi–Channel Speech Enhancement 121
Djamila Mahmoudi and Andrzej Drygajlo

The auditory critical bands interpreted as a local kautz transformation 125
A.C. den Brinker

A Classification of Byzantine Singing Voices Based on Singer's Formant 129
D.S. Delviniotis

TM P–7
Transforms and Signal Compression

Some thoughts on Morphological Pyramids and Wavelets 133
Henk J.A.M. Heijmans and John Goutsias

New Fast DCT algorithms for composite sequence lengths 137
Guoan Bi and L.W. Yu

Fast Algorithms for the Recursive Computation of Two–Dimensional Discrete Cosine Transform 141
Wen–Hsien Fang, Neng–Chung Hu and Shih–Kuo Shih

Fast Actualization of Moments in Sliding Window Applications 145
Judit Martínez and Federico Thomas

The Fast Computation of DCT in JPEG Algorithm 149
Miodrag Popović and Tomislav Stojić

Optimized Frame Design for a Matching pursuit Based Compression Scheme 153
Kjersti Engan, John Håkon Husøy and Sven Ole Aase

Transform–Domain Polynomial Filtering 157
Gábor Péceli, Tamás Kovácsházy and Annamária R. Várkonyi–Kóczy

Design of Multi–Delay Predictive Filters Using Dynamic Programming 161
Konsta Koppinen, Jari Yli–Hietanen and Peter Händel

Google Exhibit No. 1004
Page 016

Google Exhibit No. 1063
Page 0217

Signal Segmentation Using Time–Scale Signal Analysis 165
Aleš Procházka, Magdaléna Kolínová and Jaroslav Stříbrský

Anytime Algorithms in Embedded Signal Processing Systems 169
Annamária R. Várkonyi–Kóczy and Tamás Kovácsházy

Fast Algorithms for Reduction a Modulo Polynomial and Vandermonde Transform Using FFT 173
Alexander M. Krot and Helena B. Minervina

TA S–1
Array Signal Processing

Robust Adaptive Beamforming: Experimental Results 177
A.B Gershman, E. Németh and J.F. Böhme

Modified Subspace Smoothing Technique for Multipath Direction Finding 181
Mei Feng and Karl–Dirk Kammeyer

Self Calibration in Presence of correlated Discrete Sources 185
O. Grellier and P. Comon

On forward–backward mode for direction of arrival estimation 189
Magnus Jansson and Petre Stoica

MVDR beamforming with inverse updating 193
Marc Moonen and Ian K. Proudler

High Resolution Array Processing for non circular signals 197
C. Adnet, P. Gounon and J. Galy

Spatio–Temporal Coding for Radar Array Processing 201
Philippe Calvary and Denis Janer

A method of direction finding operating on a heterogeneous array 205
Yvon Erhel and Louis Bertel

Adaptive Separation of Sources and Estimation of their Directions of Arrival 209
Odile Macchi and Zied Malouche

Maximum Likelihood Time–of–Arrival Estimation using Antenna Arrays:
Application to Global Navigation Satellite Systems 213
Gonzalo Seco and Juan A. Fernández–Rubio

Simultaneously Estimating Frequency and direction of Coherent Signal
via an Array Triplet in Motion 217
Xin Su and Tianqui Chen

TA SS–2
Video Compression and Communications

Error Resilience and Concealment in Video Coding 221
[Invited Lecture]
A.K. Katsaggelos, F. Ishtiaq, L.P. Kondi, M.–C. Hong, M. Banham and J. Brailean

Networking Technologies for Future Video Communications 229
Koichi Sano, Sadayasu Ono and Naohisa Ohta

Multiple Description Coding Using Non–Hierarchical Signal Decomposition 233
Yao Wang

Rate control for delay–constrained video transmission 237
Chi–Yuan Hsu and Antonio Ortega

Compression and Packetization for MPEG–2/4 Video and Audio 241
Tokumichi Murakami, Kohtaro Asai and Hideo Ohira

MPEG–4 over Wireless Network 245
Ralf Schaefer

Optimization of Wireframe Model Adaptation and Motion Estimation in a Rate–Distortion Framework 249
Dimitrios Tzovaras, Stavros Vachtsevanos and Michael G. Strintzis

High–Order Motion Compensation for Low Bit–Rate Video 253
Bernd Girod, Thomas Wiegand, Eckehard Steinbach, Markus Flierl and Xiaozheng Zhang

A Contribution to Video Coding Using Layers 257
L. Torres and M. Lecha

TA S–3a
Multirate Systems

Results on Cyclic Signal Processing Systems 261
P.P. Vaidyanathan

Non–uniformly subsampled filter banks 265
B.E. Sarroukh and A.C. den Brinker

A Design Method of 2–D Axial–Symmetric Paraunitary Filter Banks with a Lattice Structure 269
Shogo Muramatsu, Akihiko Yamada and Hitoshi Kiya

Versatile Building Blocks for Multirate Processing of Bandpass Signals 273
Markku Renfors and Teemu Kupiainen

Fast filter bank synthesis by periodical anamorphosis 277
C. Croll, D. Pellerin and J. Hérault

TA S–3b
Filter Design

Constrained Least Squares Design of FIR Filters Using Iterative Reweighted Least Squares 281
C.S. Burrus

Design of polynomial interpolation filters based on Taylor series 283
Jussi Vesma, Ridha Hamila, Tapio Samaräki and Markku Renfors

A Design Method for Half–Band Filters 287
A.N. Wilson, Jr. and H.J. Orchard

Google Exhibit No. 1004
Page 017

Google Exhibit No. 1063
Page 0218

Design of multiplierless elliptic IIR halfband filters and Hilbert transformers 291
Miroslav D. Lutovac and Ljiljana D. Milić

New Realizations of Second–Order All–Pass Transfer Functions 295
Ana D. Simić and Miroslav D. Lutovac

TA S–4a

Face Recognition

Linear projection algorithms and morphological dynamic link architecture for frontal face verification 299
Constantine Kotropoulos and Ioannis Pitas

Nonlinear scale decomposition based features for visual speech recognition 303
Iain Matthews, J. Andrew Bangham, Richard Harvey and Stephen Cox

Wire Frame Fitting For Automatic Tracking In Model–Based Video Coding 307
P.M. Antoszczyszyn, J.M. Hannah and P.M. Grant

Region–based segmentation and tracking of human faces 311
Verónica Vilaplana, Ferran Marqués, Philippe Salembier and Luis Garrido

Facial Feature Extraction in Frontal Views Using Biometric Analogies 315
Sofia Tsekeridou and Ioannis Pitas

TA S–4b

Image Restoration

The Dual–Tree Complex Wavelet Transform: A New Efficient Tool for Image Restoration and Enhancement 319
Nick Kingsbury

Spectral Ampitude Estimation–based X–Ray Image restoration: an extension of a speech enhancement approach 323
Til Aach and Dietmar Kunz

Bayesian deconvolution of Poissonian point sources 327
Guillaume Stawinski, Arnaud Doucet, Patrick Duvaut

Modelling PSF of Scanning Electron Microscopes for image Restoration 331
J. Swindells, M. Razaz and K. Tovey

Iterative Blind Image Restoration Using Local Constraints 335
Kaaren May, Tania Stathaki and Aggelos Katsaggelos

TA S–5

Speech Recognition

On the Integration of Dialect and Speaker Adaptation in a Multi–Dialect Speech Recognition System 339
V. Digalakis, V. Doumpiotis and S. Tsakalidis

Combining Bayesian Learning and Vector Field Smoothing for On–Line Incremental Speaker Adaptation 343
C. Vair and L. Fissore

Identification of Spoken European Languages 347
Diamantino Caseiro and Isabel Trancoso

A Bayesian Triphone Model with Parameter Tying 351
Ji Ming, Marie Owens and F. Jack Smith

Spectral Subtraction and Missing Feature Modeling for Speaker Verification 355
Andrzej Drygajlo and Mounir El–Maliki

Blind Equalization in the Cepstral Domain for Robust Telephone Based Speech Recognition 359
Laurent Mauuary

Robust Speech Recognition Algorithms in A Car Noise Environment 363
Lin Cong and Saf Asghar

Frame Pruning for Speaker Recognition 367
L. Besacier and J.F. Bonastre

Integration of Parsing and Incremental Speech Recognition 371
Sven Wachsmuth, Gernot A. Fink and Gerhard Sagerer

Optimizing Hidden Markov Models for Chinese An–set Syllables 375
Q.H. He and S. Kwong

TA P–6

Pattern Recognition and Neural Networks

A unified approach to laterally–connected neural nets 379
Simone Fiori and Aurelio Uncini

A Class of Fast Complex Domain Neural Networks for Signal Processing Applications 383
Aurelio Uncini and Francesco Piazza

Efficient Layer–Wise Learning of a Feedforward Neural Networks Using the Backpropagation 387
Nikolai S. Rubanov

Recurrent Neural Networks and Filters Adaptation with Stability Control 391
Paolo Campolucci and Francesco Piazza

A Nonexclusive Classification System Based on Co–operative Fuzzy Clustering 395
F.M. Frattale Mascioli, G. Risi, A. Rizzi and G. Martinelli

Fuzzy Logic control Based on Weighted Rules using Neural Network 399
Jae–Soo Cho and Dong–Jo Park

The edge detection from images corrupted by mixed noise using fuzzy rules 403
Hiroaki Ishii, Akira Taguchi and Mototaka Sone

Google Exhibit No. 1004
Page 018

Google Exhibit No. 1063
Page 0219

A Combination of Classical and Fuzzy Classification Techniques on a Self Organized Memories (SOM).–Type Neural Network Computational Platform 407
Spyros G. Tzafestas and Sotiris N. Raptis

A New Decision Criterion for Feature Selection. Application to the Classification of Non Destructive Testing Signatures 411
Latifa Oukhellou, Patrice Aknin, Hervé Stoppiglia and Gérard Dreyfus

A Quantitative Matching Technique Based on Eigenvector Approach 415
Sang Ho Park, Kyoung Mu Lee and Sang Uk Lee

Direct Evaluation of Frame–Based Nonstationary Pattern Recognition Methods by Using Bhattacharyya Distance 419
Milan Marković, Milan Milosavljević and Branko Kovačević

Handwritten Farsi Character Recognition Using Evolutionary Fuzzy Clustering 423
Mehdi Dehghan and Karim Faez

Hidden Markov Model with Nonstationary States 427
Mohamed Ali Mahjou and Noureddine Ellouze

Experimental Car Plate Recognition System by Neural Netwroks and Image Processing 431
R. Parisi, E.D. Di Claudio, G. Lucarelli and G. Orlandi

Parameter Estimation of Conics: Application to handwritten digits 435
Mounir Amara, Pierre Courtellemont and Denis de Brucq

Multisensor Object Identification on Airports using Autoassociative Neural Networks 439
Karin Haese

Neural Network–Based Event Detection for Surveillance Applications 443
Ulf Knoblich, Michael Trompf, Siegfried Bär and Benoit Budiscak

Noise Elimination in Approximation and Time Series Prediction with Hinging Hyperplanes 447
S.R. Baldomir and D. Docampo

Multiscale versus Multiresolution Analysis for Multisensor Image Fusion 451
Youcef Chibani and Amrane Houacine

Randomized Regression in Differentiators 455
Sari Siren and Pauli Kuosmanen

TA P–7
Architectures I

A Study on Discrete Wavelet Transform Implementation for a High Level Synthesis Tool 459
Jean–Marc Tourreilles, Christophe Nouët and Eric Martin

A 500 MHz 2d–DWT VLSI Processor 463
A. Brizio, G. Masera, G. Piccinini, M. Ruo Roch and M. Zamboni

Improving "Performance vs. Silicon Size" Tradeoffs Using Coprocessors. 467
Alain Pegatoquet, Michel Auguin and Emmanuel Gresset

Single Chip DSP Array Processor: 100 Million+ Transistors with Multithreading Approach 471
Radovan Sernec, Matej Zajc and Jurij F. Tasić

Asynchronous Timing Model for High–Level Synthesis of DSP Applications 475
Okito Dedou, Daniel Chillet and Olivier Sentieys

DSP Architecture for real time digital video converter 479
Luisa Franchina

A PRML Equalizer for Hard Disk Drives with Low Sensitivity to Sampling Phase Variation 483
A. Gerosa and G.A. Mian

A Recursive Algorithm for the Generation of Space–Filling Curves 487
Greg Breinholt and Christoph Schierz

Automated Parallel–Pipeline Structure of FFT Hardware Design for Real–Time Multidimensional Signal Processing 491
A. Petrovsky and M. Kachinsky

Parallel Implementation of a Face Location Algorithm Based on the Hough Transformation 495
F. Yang, E. Drege, M. Paindavoine and H. Abdi

GSLC Architecture for sequence detectors using spatial diversity 499
Miguel A. Lagunas, Ana I. Perez–Neira and J. Vidal

A pipelined architecture for DLMS algorithm considering both hardware complexity and output latency 503
Tadaaki Kimijima, Kiyoshi Nishikawa and Hitoshi Kiya

A G–transform based systolic array for Least Squares Problems 507
Evangelos Zervas, Athanassia Alonistioti and Nikos Passas

VLSI Implementation and Complexity Comparison of Residue Generators Modulo 3 511
Stanislaw J. Piestrak, Fabrice Pedron and Olivier Sentieys

Adaptative filters implementation performances under power dissipation constraint 515
Stéphane Gailhard, Nathalie Julien and Eric Martin

Behavioral Synthesis of Digital Filters Using Attribute Grammars 519
George Economakos, George Papakonstantinou and Panayiotis Tsanakas

The Impact of Data Characteristics on Hardware Selection for Low–Power DSP 523
G. Keane, J.R. Spanier and R. Woods

Google Exhibit No. 1004
Page 019

Google Exhibit No. 1063
Page 0220

Volume II

PLENARY TALK

Partial Differential Equations in Image Analysis: Continuous Modeling, Discrete Processing 527
Petros Maragos

WM SS–1
Nonlinear Signal Processing for Communications

Nonlinear Signal Processing for Adaptive Equalisation and Multi–User Detection 537
[Invited Lecture]
Bernard Mulgrew

Channel Equalization For Coded Signals In Hostile Environments 545
Kristina Georgoulakis and Sergios Theodoridis

A Combined LMS–SOM Algorithm for time–varying non–linear channel equalization 549
Steven Bouchired, Mohamed Ibnkahla and Williams Paquier

Non–Linear Equalizers that Estimate Error Rates During Reception 553
Jesus Cid–Sueiro, Anibal R. Figueiras–Vidal

On the Nonlinearity of Linear Prediction 557
Gernot Kubin

A Likelihood Framework for nonlinear Signal Processing with Finite normal Mixtures 561
Tülay Adali, Bo Wang, Xiao Liu and Jianhua Xuan

Analysis of a Nonlinear Equalizer Based on High–Order Statistics 565
J.B. Destro Filho, G. Favier and J.M. Travassos Romano

Bit Error Rate Optimization of DS–CDMA Receivers 569
Ioannis N. Psaromiligkos and Stella N. Batalama

Constrained Pulse Shape Synthesis for Digital Communications 573
P.L. Combettes and P. Bondon

WM S–2
Shape Analysis and Recognition

Sequential Local Transform Algorithms for Gray – Level Distance Transforms 577
Pekka J. Toivanen and Hesham Elmongui

Hierarchical Skeleton Extraction based on a Deformable Particle System 581
Franck Angella, Olivier Lavialle and Pierre Baylou

Shape representation for object correspondence based on sub–graph matching and Fourier descriptors 585
Ferran Marqués and Guillermo Gutiérrez

Rapid location of convex objects in digital images 589
E.R. Davies

Boundary Tracking in 3D Binary Images to Produce Rhombic Faces for A Dodecahedral Model 593
Edgar Garduño, Gabor T. Herman and Hava Katz

Efficient Curvature–based Shape Representation for Similarity Retrieval 597
Farzin Mokhtarian, Sadegh Abbasi and Josef Kittler

Model Acquisition and Matching in Tagged Object Recognition (TOR) 601
L.M. Soh, J. Matas and J. Kittler

Form Identification and Skew Detection from Projections 605
N. Liolios, N. Fakotakis and G. Kokkinakis

The use of steerable filters for hand–written character recognition 609
Emir Tufan and Vedat Tavsanoglu

Stereo–vision via connected–set operators 613
Richard Harvey, Kimberly Moravec and J. Andrew Bangham

WM S–3
Time – Frequency Analysis

Optimum oversampling in the rectangular Gabor scheme 617
Martin J. Bastiaans

Half–Quadratic Regularization of Time–Frequency AR Analysis for Recovery of Abrupt Spectral Discontinuities and their Detection by a Recursive Siegel Metric based on Information Geometry 621
Frédéric Barbaresco

Representation of Pseudoperiodic Signals by means of Pitch–Synchronous Frequency Warped Wavelet Transform 625
Sergio Cavaliere and Gianpaolo Evangelista

Algorithm for the Instantaneous Frequency Estimation Using Time–Frequency Distributions with Adaptive Window Width 629
Ljubiša Stanković and Vladimir Katkovnik

Tracking of spectral lines in an ARCAP time–frequency representation 633
M. Davy, B. Leprettre, C. Doncarli and N. Martin

Detection and tracking of multi–periodic signals 637
I.J. Clarke and G. Spence

Sound Signature Analysis using Time–Frequency Signal Processing: Application to Active Stall Avoidance in Axial Compressors 641
Thuyen Le, Thomas Dombek and Manfred Glesner

Google Exhibit No. 1004
Page 020

Google Exhibit No. 1063
Page 0221

Extended Wavelet Transforms in Acoustic Diagnosis 645
Georg Wirth and Dieter A. Mlynski

Quantization Effects in Implementation of Distributions from the Cohen Class 649
Veselin Ivanović, Ljubiša Stanković and Zdravko Uskoković

Cross–Terms Free Forms of Some Quadratic and Higher Order Time–Frequency Representations 653
Ljubiša Stanković and Srdjan Stanković

WM S–4a
Biomedical Signal Processing

A Nonlinear Dynamical Model for Compression and Detection of ECG Data 657
T. Schimming, M. Ogorzalek and H. Dedieu

Analysis of Brain Electroencephalograms with the Instantaneous Maximum Entropy Method 661
Yumi Takizawa, Sunao Uchida, Makio Ishiguro and Atsushi Fukasawa

Fast optimal beam determination for conformation radiotherapy treatment planning 665
Yong Yuan, William Sandham, Tariq Durrani and Charles Deehan

Single Channel Analysis of Sleep EEG: An Adaptive Database Method 669
Christian Berthomier, Jacques Prado and Odile Benoit

Blood glucose prediction for diabetes therapy using a recurrent artificial neural network 673
William Sandham, Dimitra Nikoletou, David Hamilton, Ken Paterson, Alan Japp and Catriona MacGregor

WM S–4b
Biomedical Image Processing

4–D reconstruction of the left ventricle from a single cycle ultrasound acquisition 677
Claudia Bonciu, Rodolphe Weber and Long Dang Nguyen

Ultrasonic Array Imaging Using CDMA Techniques 681
Yannis S. Avrithis, Anastasios N. Delopoulos and Grigorios C. Papageorgiou

A Comparison of several interpolation methods in 3D x–ray cone beam reconstruction 685
J.G. Donaire and I. García

Weighting Hyperparameters for 3D Bayesian estimation in Eddy Current Tomography 689
Olivier Venard and Denis Prémel

Enhancement of Mammographic Images for Detection of Microcalcifications 693
Damir Seršic and Sven Lončarić

WM S–5a
Speech Analysis

Voice Source Parameters for Speaker Verification 697
Andreas Neocleous and Patrick A. Naylor

A Nonlinear Algorithm for Epoch marking in Speech Signals Using Poincaré Maps 701
Iain Mann and Steve McLaughlin

A Geometric Algorithm for Voice Activity Detection in Nonstationary Gaussian Noise 705
Hamza Özer and S. Gökhun Tanyer

Comparison of Some Time–Frequency Analysis Methods for Classification of Plosives 709
Ewa Lukasik, Stefan Grocholewski

Speech Parameters Vector Based on Arithmetic Fourier Transform 713
Evgeny I. Bovbel and Igor E. Kheidorov

WM S–5b
Speech and Audio Coding / Compression

CELP with Priority to Critical Segments 717
Lúcio Martins da Silva and Abraham Alcaim

On the Combination of Redundant and Zero–Redundant Channel Error Detection in CELP Speech–Coding 721
Norbert Görtz

A new intraframe LSP interpolation technique for low bit rate speeh coding 725
J.S. Mao, S.C. Chan and K.L. Ho

Backward Adaptive Warped Lattice for Wideband Stereo Coding 729
Aki Härma, Unto K. Laine and Matti Karjalainen

An Algorithm for Wideband Audio Coding based on LMS / RLS Switched Predictors 733
Enzo Mumolo, Mauro Scagnol and Alberto Carini

WM P–6
Efficient Algorithms for Filter Design and Signal Reconstruction

Novel Adaptive Algorithm Based on Least Mean p–Power Error Criterion for Fourier Analysis in Additive Noise 741
Yegui Xiao and Katsunori Shida

The Summational Projection Alhgorithm for the Adaptive Volterra Filter 745
Yoshinobu Kajikawa and Yasuo Nomura

Convergence Analysis and Fast Algorithms of Volterra Adaptive Filters 749
Jinhui Chao, Atsushi Inomata and Shinpei Uno

Evolving Complex Adaptive IIR Structures 753
Seelan Sundaralingam and Ken Sharman

Fractionally–Spaced Equalization for Time–Varying Channels 757
M.–L. Alberi and I. Fijalkow

Adaptive blind separation of convolved sources based on maximization of the generalized instantaneous energy 761
Ivica Kopriva

Google Exhibit No. 1004
Page 021

Google Exhibit No. 1063
Page 0222

Adaptive Blind Equalization for Asynchronous DS–CDMA Systems Based on RLS 765
Geert Leus and Marc Moonen

Improved Neural Network Equalization by the use of Maximum Covariance Weight Initialization 769
Arto Kantsila, Mikko Lehtokangas and Jukka Saarinen

Adaptive Nonlinear Filtering with the Support Vector Method 773
Davide Mattera, Francesco Palmieri and Simon Haykin

Adaptive Weighted Vector Median Filter Using a Gradient Algorithm 777
Laurent Lucat and Pierre Siohan

Smoothing of Noisy AR Signals Using an Adaptive Kalman Filter 781
Gerhard Doblinger

Accurate LDA – spectra by resampling and ARMA – modeling 785
S. de Waele and P.M.T. Broersen

Sintrack Analysis. Application to Detection and Estimation of Flutter for Flexible Structures 789
Matthieu Jeanneau, Philippe Mouyon and Céline Pendaries

WM P–7a
Image Filtering

Nonlinear filtering of MR images using geometrically and statistically controlled diffusion 793
Ivan Bajla, Viktor Witkovský and Milan Hanajík

A novel Algorithm for Digital Image Processing Applications 797
S. Boussakta

Stable nonlinear filters with spatial prediction 801
Jamal Abbas and M. Domański

From continuum model to a detailed discrete theory of median shifts 805
E.R. Davies

Fuzzy Colour Filtering Using Relative Entropy 809
A. Fotinos, N. Laskaris and S. Fotopoulos

Decomposition and Order Statistics in Filtering 813
Dinu Coltuc and Philippe Bolon

A True order recursive algorithm for two–dimensional least squares error linear prediction and filtering 817
George–Othon Glentis

One–Dimensional Scale–Space Preserving Filters 821
Richard Harvey, Alison Bosson and J. Andrew Bangham

Transfer Function Models for Continuous and Discrete Multidimensional Systems 825
R. Rabenstein

WM P–7b
Image Restoration and Enhancement

The Periodic Step Gradient Descent Algorithm – General Analysis and Application to the Super Resolution Reconstruction Problem 829
Tamir Sagi, Arie Feuer and Michael Elad

Reconstruction of locally homogeneous images 833
Mila Nikolova

Statistical Restoration of Images Using a Hybrid Bayesian Approach 837
D. Hudson and M. Razaz

A New Approach for Restoration of NMR Signals 841
M. Razaz, R.A. Lee, P.S. Belton and K.M. Wright

Enhancing Handwritten Character Images thanks to a Re–Sampling Process Based on Convex Hull Extraction 845
B. Gosselin

A Sequential Projections Based Postprocessor for Block–based Image Coding 849
Joon–Ho Chang, Sang Hwa Lee, Jeong–Kwon Kim and Choong Woong Lee

Enhancement of Sketch Contours on Paintings Infrared Photographies: A Comparison 853
Arnaldo de Albuquerque Araújo, Renato Moreira Hadad and Beatriz V. Coelh

An Integrated system for Object Tracking and Progressive Coding Based on Statistical Morphology 857
Carlo S. Regazzoni, Andrea Teschioni and GianLuca Foresti

An inverse Problem: Histogram Equalization 861
Dinu Coltuc and Philippe Bolon

WA S–1
Architectures II

Rapid Design of Discrete Transform Cores 865
Jill K. Hunter and John V. McCanny

Low Power Implementation of Discrete Wavelet Transform 869
K. Masselos, P. Merakos, T. Stouraitis and C.E. Goutis

Efficient Implementation of the Row–Column 8x8 IDCT on VLIW Architectures 873
Rizos Sakellariou, Christine Eisenbeis and Peter Knijnenburg

Relaxed Look–Ahead Technique for Pipelined Implemantation of Adaptive Multiple–antenna CDMA Mobile Receivers 877
Ramin Baghaie, Stefan Werner and Timo Laakso

A Scheme for the VLSI Implementation of FIR Digital Filters with Reduced Latency 881
Christos Gr. Caraiscos and Kiamal Z. Pekmestzi

Efficient Bit–Level Design of an On–Board Digital TV Demultiplexer 885
S. Calvo, J. Sala, A. Pagès and G. Vàzquez

Google Exhibit No. 1004
Page 022

Google Exhibit No. 1063
Page 0223

A Distributed Adaptive Block Matching Algorithm: DIS–ABMA 889
F. Vermaut, Y. Deville, B .Macq and X. Marichal

Prediction of Decoding Time for MPEG–4 Video 893
Marco Mattavelli and Sylvain Brunetton

Real time image rotation using B–spline interpolation on FPGA's board 897
C. Berthaud, E. Bourennane, M. Paindavoine and C. Milan

Fully–Digital Highly–Modular Architecture of a Hamming–like Neural Network 901
A. Pavlidis, A. Arapoyianni and D. Loukas

WA S–2
Motion

A Fast Two Stage Translational and Warping Motion Compensation Scheme 905
D.B. Bradshaw and N.G. Kingsbury

A fast full search block matching algorithm using subblocks 909
Michael Brünig and Wolfgang Niehsen

A New Technique for Motion Estimation and Compensation of the Wavelet Detail Images 913
G. Van der Auwera, A. Munteanu, G. Lafruit and J. Cornelis

New closed–form solutions to image flow equations for ego–motion applications considering variations of the focal distance 917
José M. Menéndrez, Narciso García, Luis Salgado and Enrique Rendón

Motion segmentation and tracking using a seeded region growing method 921
I. Grinias and G. Tziritas

Motion estimation based on affine moment invariants 925
G. Tzanetakis, M. Traka and G. Tziritas

A Pseudo 3D Motion Estimator for Moving Object Estimation in Noisy Video Sequences 929
Christopher L. Topping and Jonathon A. Chambers

An Object–Based Hierarchical Motion compensation Technique Using the Greedy Method 933
Jun Seo Lee, Rin Chul Kim and Sang Uk Lee

Video Object Manipulation based on Mosaic Representation for post–production applications 937
H. Nicolas

Tracking Markers for Human Motion Analysis 941
Pascual J. Figueroa, Neucimar J. Leite, Ricardo L. Barros and René Brenzikofer

WA SS–3
Acoustic Echo and Noise Control

Acoustic Echo and noise Control – A Long Lasting Challenge 945
[Invited Lecture]
Pia Dreiseitel, Eberhard Hänsler and Henning Puder

Speech Quality Evaluation of Hands–Free Telephones During Double Talk: New Evaluation Methodologies 953
H.W. Gierlich, F. Kettler and E. Diedrich

Combined systems for noise reduction and echo cancellation 957
Christophe Beaugeant and Pascal Scalart

Combined Residual Echo and noise Reduction: A Novel Psychoacoustically Motivated Algorithm 961
Stefan Gustafsson and Peter Jax

Speech Enhancement for Mobile Telephony Based on Non–uniformly Spaced Frequency Resolution 965
Pia Dreiseitel and Henning Puder

An Acoustic Echo Canceller with Compensation of Nonlinearities 969
Alexander Stenger and Rudolf Rabenstein

On the use of new adaptive subband structures in acoustic echo cancellation 973
Mariane R. Petraglia, Rogerio G. Alves and Antonio Petraglia

Steady – State Solutions of the Extended LMS Algorithm for Stereophonic Acoustic Echo Cancellation with Leakage or Signal Conditioning 977
Tetsuya Hoya, Jonathon A. Chambers, Neil Forsyth and Patrick A. Naylor

DSP Implementation and Performance Evaluation of a Stereo Echo Canceler with Pre–Processing 981
Yann Joncour, Akihiko Sugiyama and Akihiro Hirano

WA S–4a
Statistical Signal Processing

A General Maximum Likelihood Classifier for Modulation Classification 985
C. Le Martret and D. Boiteau

Approximation of a–Stable Probability Densities Using Finite Gaussian Mixtures 989
Ercan E. Kuruoğlu, Christophe Molina and William J. Fitzgerald

Wavelet Thresholding for a Wide Class of noise Distributions 993
D. Leporini and J.–C. Pesquet

Information Criteria Based Edge Detection 997
Frédéric Jouzel, Christian Olivier and Abdelaziz El Matouat

Estimating the Predictability and the Linearity of a Process by Kernels 1001
Andreas Poncet and George S. Moschytz

WA S–4b
Parameter Estimation

Multi–user Channel Estimation Exploiting Pulse Shaping Information 1005
Erik Lindskog and Jonas Strandell

A Subspace Fitting–like Method for Almost Low Rank Models 1009
Mats Bengtsson

Google Exhibit No. 1004
Page 023

Google Exhibit No. 1063
Page 0224

MAP Based Schemes for Detection of Abrupt Changes for Fading Channels 1013
Catharina Carlemalm and Fredrik Gustafsson

Separate Temporal and Spatial Parametric Channel Estimation 1017
Jonas Strandell and Erik Lindskog

Parameter estimation in partitioned nonlinear stochastic models 1021
Ola Markusson and Håkan Hjalmarsson

Source Localization in Shallow Water in The Presence of Sensor Depth Uncertainty 1025
Assi Jakoby, Jason Goldberg and Hagit Messer

WA S–5a
Image Enhancement

Reconstructing Missing Regions in Colour Images Using Multichannel Median Models 1029
Steven Armstrong, Anil C. Kokaram and Peter J.W. Rayner

Unsharp Masking–Based Approach for Color Image Processing 1033
Faouzi Alaya Cheikh, Lazhar Khriji and Moncef Gabbouj

The role of Gamma correction in colour image processing 1037
Wilfried Kubinger, Markus Vincze and Minu Ayromlou

Quality limits in color printing with more than three primary colors 1041
Werner Praefcke

Image Halftoning Using Optimized Dot Diffusion 1045
Murat Meşe and P.P. Vaidyanathan

WA SS–5b
Image Quality Evaluation

A Technique for Image Quality Assessment based on a Human Visual System Model 1049
Wilfried Osberger, Anthony Maeder and Neil Bergmann

Subjective Measure of Edge degradation for Vector Quantized Color Images 1053
Christophe Charrier and Hocine Cherifi

Towards a Visual Quality Metric for Digital Video 1057
Andrew B. Watson

Psychovisual measurement and Distortion Metrics for Image Sequences 1061
Edmund M. Yeh, Anil C. Kokaram and Nick G. Kingsbury

Monitoring the Quality of MPEG Coded Video 1065
K.T. Tan and M. Ghanbari

A Human Vision System Model for Objective Image Fidelity and Target Detectability Measurements 1069
Jeffrey Lubin

WA P–6
3–D Imaging

Modeling 3D Textured Objects by Fusion of multiple Views 1073
Timothée Jost, Christian Schütz and Heinz Hügli

Scene Analysis Using Fusion of Range and Color Data 1077
Philippe Pujas and Marie–José Aldon

Tracking Camera Calibration in Multi–Camera Sequences through Automatic Feature Detection and Matching 1081
Federico Pedersini, Augusto Sarti and Stefano Tubaro

Range Image Registration for Free–Form Surfaces Using Curvature Information as Invariance 1085
Markus Rieder, Nicolas Guil and R. Dillmann

Visualisation of Videoconference Image Sequences Using VRML 2.0 1089
Ioannis Kompatsiaris and Michael G. Strintzis

Object Segmentation in 3D Images Based on Alpha–Trimmed Mean Radial Basis Function Network 1093
Adrian G. Borş and Ioannis Pitas

Classification of Archaeological Fragments Using a Description Language 1097
Robert Sablatnig, Christian Menard and Walter Kropatsch

On the Application of Light field Reconstruction for Statistical Object Recognition 1101
B. Heigl, J. Denzler and H. Niemann

Virtual Environment Generation by CAD–Based Methodology for Underwater Vehicle Navigation 1105
Leila De Floriani, Vittorio Murino, Goffredo G. Pieroni and Enrico Puppo

Combining Intensity and Stereo Data to Improve Satellite Urban Scenes Modeling 1109
N. Paparoditis and M. Cord

WA P–7a
Applications and Architectures

A new FPGA architecture for image processing: CYCLOP 1113
Philippe Guermeur

Multi – FPGA Processor for gray–scale Morphology 1117
Mohamed Akil and Shahram Zahirazami

Radon Transform for Internal Wave Detection and Orientation 1121
Josep A. Ródenas, Nicolas Mandelert and René Garello

Application of Momentary fourier Transform to the SPECAN SAR Processing 1125
Sandor Albrecht and Ian Cumming

Synthetic Aperture Radar Interferometry Using Ground Slope Vector to Phase Unwrapping 1129
Xavier Dupuis, Pierre Mathieu and Michel Barlaud

Google Exhibit No. 1004
Page 024

Google Exhibit No. 1063
Page 0225

WA P-7b

Speech Processing

Automatic pitch marking for speech transformations via TD–PSOLA 1133
Yves Laprie and Vincent Colotte

The method of pitch frequency detection on the base of tuning to its harmonics 1137
V. Sercov and A. Petrovsky

Acoustic Measure of Noise Energy in Vocal Folds Operated Patients 1141
C. Manfredi, M.D' Aniello and P. Bruscaglioni

Analysis of Pitch–Synchronous Modulation Effects by using Analytic Filters 1145
Unto K. Laine

Perceptual Evidence for FO Variability Constraints of Phrase–Initial Accents in Greek 1149
Christos Malliopoulos and George Carayannis

The Impact of relative average duration of vowels in Greek on the perception of a constant synthetic speech rate 1153
Stavroula–Evita F. Fotinea, Michael A. Vlahakis and George V. Carayannis

Text–to–Speech Synthesis in Slovenian Language 1157
Tomaž Šef, Aleš Dobnikar and Matjaž Gams

Comparison of Two Different Text–to–speech Alignment systems: Speech Synthesis based VS. Hybrid HMM/ANN 1161
O. Deroo, F. Malfrere and T. Dutoit

A Fuzzy Approach to Text–to–Speech Synthesis 1165
Enzo Mumolo and William Costanzo

A method to choose an appropriate concatenating position for automatically generated synthesis unites 1169
Hun Jae Park, Sang Hun Kim, Min Soo Han and Jae Ho Chung

A high performance vowel spotting system based on a multi–stage architecture 1173
J. Sirigos, N. Fakotakis and G. Kokkinakis

On a Time–varying Complex Speech Analysis 1177
Keiichi Funaki, Yoshikazu Miyanaga and Koji Tochinai

Autocorrelation Analysis of Speech Signals Using the Number Theoretic Transforms over the Direct Sums of Finite Fields 1181
Nikolai S. Rubanov and Eugene I. Bovbel

Speech Analysis and Synthesis Using Instantaneous Amplitudes 1185
Gang Li and Lunji Qiu

Spectral Estimation of Voiced Speech with Regressive Linear Prediction 1189
Susanna Varho and Paavo Alku

The performance of a Novel Pre–processor for the Learning Vector Quantisation Classifier 1193
H. Kelleher and E. Chilton

Low Delay Phone Recognition 1197
José A.R. Fonollosa, Eloi Battle and José B. Mariño

Design of a Command Interface with a Dynamic Grammar Speech Recognition Engine 1201
S. Kruger, S. Chennoukh, J. Flanagan and L. Rothkrantz

ADPCM with Nonlinear Prediction 1205
Marcos Faúndez–Zanuy and Oscar Oliva–Suarez

Non–Linear Processing in Cochlear Space Sub–Bands using Artificial Neural Networks for Multi–Microphone Adaptive Speech Enhancement 1209
Amir Hussain and Douglas R. Campbell

Google Exhibit No. 1004
Page 025

Google Exhibit No. 1063
Page 0226

Volume III

PLENARY TALK

Optimazation of filter banks based on properties of the input signal 1213
P.P. Vaidyanathan

ThM S–1
Adaptive Systems I

A Block Adaptive DFE in the Frequency Domain based on Tentative Decisions 1221
Kostas Berberidis and Panos Karaivazoglou

Implementation of a block adaptive filter working in the frequency domain combined with a robust adaptation control 1225
Bernhard Nitsch

A Fast Prewhitened Affine Projection Algorithm 1229
Karim Maouche and Dirk T.M. Slock

A Fast Multichannel Adaptive Filtering Algorithm 1233
Mounir Bhouri, Mamadou Mboup and Madeleine Bonnet

Simplified FLS Algorithm for Linear Phase Adaptive Filtering 1237
Leonardo S. Resende, João Marcos, T. Romano and Maurice G. Bellanger

A Delayless Subband Adaptive Filter Using the Projection Method 1241
Hideaki Sakai and Shigeyuki Miyagi

Performance Limitations of Subband Adaptive Filters 1245
S. Weiß, R.W. Stewart, A. Stenger and R. Rabenstein

Step–size optimization of the BNDR–LMS Algorithm 1249
J. A. Apolinário Jr., P.S.R. Diniz, T.I. Laakso and M.L.R. de Campos

LMS Tracking Behavior Under Periodically Changing Systems 1253
Markus Rupp

A Controlled Block Implementation of the NLMS Algorithm 1257
Kensaku Fujii and Juro Ohga

ThM SS–2
3–D Imaging and Virtual Inspection of Objects and Places

Automatic 3D Model Acquisition and Generation of New Images from Video Sequences 1261
[Invited Lecture]
Andrew Fitzigbbon and Andrew Zisserman

Synthesis of virtual views of a scene from two or three real views 1269
Georges M. Quénot

3D Photography using shadows 1273
Jean–Yves Bouguet and Pietro Perona

Towards Flexible 3–D Digitizing Systems 1277
P. Hébert, J. Tremblay, F. Blais, H. Chotard and S. Dyck

Acquisition and 3D Registration of Image Sequences for Structured Light Based Free–Form Surface Reconstruction 1281
C. Schoenenberger, P. Graebling and E. Hirsch

Evaluation of the Construction of Novel Views by a combination of Basis Views 1285
Bernard Buxton, Zahid Shafi and John Gilby

Three Dimensional View Registration by a Frequency Domain Technique 1289
G.M. Cortelazzo, G. Doretto, L. Lucchese and S. Totaro

Multi–View Modeling & Synthesis 1293
Chen Brestel and Shimon Ullman

ThM S–3
Neural Networks and Fuzzy Systems

Adaptive Learning Algorithms for Semantic Object Extraction 1297
Nikolaos Doulamis, Anastasios Doulamis and Stefanos Kollias

Recurrent Neural Networks for Signal Processing Trained by a new Second Order Algorithm 1301
Paolo Campolucci, Michele Simonetti and Aurelio Uncini

Neural Networks with Hybrid Morphological / Rank / Linear Nodes and their Application to Handwritten Character Recognition 1305
Lúcio F.C. Pessoa and Petros Maragos

Decision Level Fusion by Clustering algorithms for Person Authentication 1309
Vassilios Chatzis, Adrian G. Bors and Ioannis Pitas

Supervised Multisensor Tracking algorithm 1313
Vincent Nimier

Second Order Hebbian Neural Networks and Blind Source Separation 1317
Konstantinos I. Diamantaras

Divide and conquer algorithms for constructing neural networks architectures 1321
K. Koutroumbas, A. Pouliakis and N. Kalouptsidis

Implementing Size–Optimal Discrete Neural Networks Require Analog Circuitry 1325
Valeriu Beiu

Fast Formation of Invariant Feature Maps 1329
Stephen McGlinchey and Colin Fyfe

Google Exhibit No. 1004
Page 026

Google Exhibit No. 1063
Page 0227

ThM S–4
Signal Processing for Communications I

Performance Evaluation of Adaptive Subspace Detectors Based on Stochastic Representations 1333
Shawn Kraut and Louis Scharf

Low Power Detection 1337
Mohammed Nafie and Ahmed Tewfik

MLSE and Spatio–Temporal Interference Rejection Combining with Antenna Arrays 1341
David Asztély and Björn Ottersten

Interference suppression in SS systems: a comparison between PTV P/S–type and W–type filters 1345
Giacinto Gelli

A Space/Time Pre–Equalization Technique for Down–Link Signal Transmission in Time–Division–Duplex (TDD) Mobile Multimedia Communications 1349
Shigeru Tomisato, Kazuhiko Fukawa and Tadashi Matsumoto

Fast Phase Sequences Spreading codes for CDMA using FFT 1353
E. Del Re, R. Fantacci and L.S. Ronga

A new Orthogonal Sequence for DS/CDMA Systems 1357
Seong Ill Park, Hong Gil Kim, Iickho Song, Suk Chan Kim, Yun Hee Kum and Jooshik Lee

Modelling Sea Clutter Using Conditional Heteroscedastic Models 1361
Jacek L. Noga and William J. Fitzgerald

Robust Image Restoration Matched with Adaptive Aperture Formation in Radar Imaging systems with Sparse Antenna Arrays 1365
Ivan Prudyus, Sviatoslav Voloshynovskiy and Taras Holotyak

A Real–Time Performance Evaluation Technique for Future DSP–Based "Software" Wireless Radio 1369
Lorenzo Prosperi, Daniele Gianfelici, Luisa Franchina and Savvas A. Kosmopoulos

ThM S–5
Nonlinear Signal Processing and Systems

An Efficient Order Recursive Algorithm for Volterra System Identification 1373
George–Othon Glentis and Nicholas Kalouptsidis

Signal Modeling Using Piecewise Linear Chaotic Generators 1377
Thomas Schimming, Marco Götz and Wolfgang Schwarz

Analysis of synchronization of chaotic systems based on local conditional Lyapunov exponents 1381
Zbigniew Galias

Inversion of H–ARMA Models 1385
David Declercq, Patrick Duvaut and Jerome Soubielle

A Time–Frequency Method for Nonlinear System Classification in Presence of Noise 1389
Lorenzo Galleani and Letizia Lo Presti

Nonlinear frame–like decompositions 1393
B. Pouye, A. Benazza–Benyahia, I. Pollak, J.–C. Pesquet and H. Krim

Numerical Integration of nonlinear PDEs Using MD Passive Circuits 1397
F.N. Koumboulis and B.G. Mertzios

Digital Compression of Analytic Signals Using Nth Root 1401
M. Bellanger and K. Ouaissa

Generation of Idempotent Monotone Boolean Functions 1405
Ilya Shmulevich

Nonlinear Decision Feedback Equalization Using RCPL Network 1409
Levent Demirekler, Tülay Adali and Xiao Liu

ThM P–6a
Speech Coding

A new hybrid CELP–Harmonics speech coder at low bit rates 1413
Laurent Hubaut, Olivier van der Vrecken and Florence Coulon

Increasing Quality of CELP Coders by Source–Filter Interrelation Using self Organizing Maps 1417
Gökhan Avkaroğullari and Tolga Çiloğlu

On the Application of a Psychoacoustically Motivated Speech–Quality Measure in CELP Speech–Coding 1421
Markus Hauenstein and Norbert Görtz

Fast LSP Calculation and Quantization with Application to the CELP FS1016 Speech Coder 1425
S. Grassi, M. Ansorge and F. Pellandini

A preliminary study of an audio–visual speech coder: using video parameters to reduce an LPC vocoder bit rate 1429
Elodie Foucher, Gang Feng and Laurent Girin

Warped Linear Predictive Audio Coding in Video Conferencing Application 1433
Kalle Palomäki, Aki Härmä and Unto K. Laine

Variable–rate speech coding: coding unvoiced frames with 400 bps 1437
Wolfram Ehnert

Performance of Discrete Fourier Transform with small overlap in Transform–Predictive–Coding–based coders 1441
A. Jbira

Improved Lost Frame Recovery Techniques for ITU–T G.723.1 Speech Coding System 1445
Grant Ho, Suat Yeldener and Marion Baraniecki

An Adaptive Quantization Using IIR Filter Bank for Speech Compression 1449
Hiroto Saito and Shogo Nakamura

Google Exhibit No. 1004
Page 027

Google Exhibit No. 1063
Page 0228

ThM P–6b
Speech Enhancement / Recognition

Spatial Coherence Exploitation which Yields Non–Stationary Noise Reduction in Subband Domain 1453
R. Atay, E. Mandridake, D. Bastard and M. Najim

Adaptive Kalman Filter for Speech Enhancement from Colored Noise 1457
M. Gabrea, E. Mandridake and M. Najim

A Robust Begin–End Point Detector for Highly noisy Conditions 1461
R. Martínez, A. Alvarez, P. Gómez, V. Nieto, V. Rodellar, M. Rubio and M. Pérez

A double talk detector based on the partial coherence function 1469
R. Le Bouquin Jeannès and G. Faucon

Speaker Normalization for Automatic Speech Recognition – An on–line Approach 1473
Ioannis Dologlou, Tom Claes, Louis ten Bosch, Dirk Van Compernolle and Hugo Van Hamme

On the Equivalence between Predictive Models for Automatic Speech Recognition 1477
Bojan Petek

Cross–Language Text–Independent Speaker Identification 1481
Geoffrey Durou and Frédéric Jauquet

A Novel Model for Phoneme Recognition using Phonetically Derived Features 1485
Naomi Harte, Saeed Vaseghi and Paul McCourt

Diverse Processing in Cochlear Spaced Sub–Bands for Multi–Microphone Adaptive Speech Enhancement in Reverberant Environments 1489
Amir Hussain, Douglas R. Campbell and T.J. Moir

ThM P–7a
Image and Video Coding

A General–Tree–Structured Vector Quantizer for Image Progressive Coding 1493
Lin Yu Tseng and Shiueng Bien Yang

Rate Distortion Optimal Contour Compression Using Cubic B–Splines 1497
J. Zaletelj, R. Pecci, F. Spaan, A. Hanjalic and R.L. Lagendijk

Coding of Arbitrarily Shaped Video Objects Using B–Spline Wavelets 1501
Mervi Sepponen and Visa Koivunen

Fast quasi–DCT algorithm for shape–adaptive DCT image coding 1505
Ryszard Stasiński and Janusz Konrad

Fast Approximation of DCT Kernel in JPEG 1509
Anton Marček and Gregor Rozinaj

Statistical Modelling of Full Frame DCT Coefficients 1513
M. Barni, F. Bartolini, A. Piva and F. Rigacci

Quality Scalable Coding of Selected Region 1517
Wook–Joong Kim, Jong–Won Yi and Seong–Dae Kim

Lossless compression of images by a Search order based entropy coding 1521
J. Jiang and C.V. Brett

A context–Based Recursive Nonlinear Interpolation for Near–Lossless Coding of X–Ray Images 1525
Bruno Aiazzi, Stefano Baronti, Franco Lotti and Luciano Alparone

A study of a lossless image compression algorithm using adaptive prediction and context–based entropy coding 1529
Guang Deng

Rate–Distortion Analysis of Nonlinear Quantisers for Video Coders 1533
Yoo–Sok Saw, Peter M. Grant and John M. Hannah

Hierarchical MPEG 2 video transmission on ADSL for a higher Quality of Service (QoS) 1537
M. Colin, M. Gharbi, M. Gazalet and C. Modlin

Fast color transformation by means of hierarchical table lookups 1541
P.L. Dragotti and A.R.P. Ragozini

ThM P–7b
Motion Estimation

Robust Tracker of Small, Fast–Moving Low–Contrast Targets 1545
D. Davies, P. Palmer and M. Mirmehdi

Accurate Motion Interpôlation by using a Region Based Motion Estimator 1549
R. Lancini, M. Ripamonti, S. Tubaro and P. Vicari

A Cooperative Top–Down / Bottom–Up Technique for Motion Field Segmentation 1553
R. Leonardi, P. Migliorati and G. Tofanicchio

Motion Estimation and Modeling in Video Sequences 1557
Ciro Cafforio, Eugenio Di Sciascio and Cataldo Guaragnella

The Impulse RETINA : a smart velocity sensor 1561
Emmanuel Marilly, Christophe Coroyer, Olga Cachard and Alain Faure

A Fast Block Matching Motion Estimation Algorithm Based on Simplex Minimisation 1565
Mohammed E. Al–Mualla, Nishan Canagarajah and David R. Bull

Motion estimation based on triads of Gabor filters: DSP board implementation 1569
A. Spinéi, D. Pellerin and J. Hérault

Hierarchical Estimation of Optical Flow Using Spectral Energy Method 1573
Takashi Koike, Katsuya Kondo and Nozomu Hamada

Visual Module integration for optical flow estimation 1577
Luigi Bedini, Andrea Cannata, Mario Ferraro, Emanuele Salerno and Anna Tonazzini

Google Exhibit No. 1004
Page 028

Google Exhibit No. 1063
Page 0229

ThA S–1a
Pattern Recognition

A Vector Based Approximation of KLT and its Application to Face Recognition 1581
Nicolas Tsapatsoulis, Vassilios Alexopoulos and Stefanos Kollias

Recognition of Rotated and Scaled Textures Using 2–D AR Modeling and the Fourier–Mellin Transform 1585
C. Cariou, O. Alata, C. Rosenberger, J.–M. Ogier and K. Chehdi

Point pattern matching using a genetic algorithm and voronoi tessellation 1589
Marius Tico and Corneliu Rusu

Knowledge Based Interpretation of Aerial Images Using Multiple Sensors 1593
R. Tönjes and C.–E. Liedtke

Automatic Generation of a VLSI Parallel Architecture for QRS Detection 1597
A. Koulouris, N. Koziris, T. Andronikos, G. Papakonstantinou and P. Tsanakas

ThA S–1b
Nonlinear Techniques for Channel Equalization

A fuzzy logic filter for coherent detection in Mobile Communication Receivers 1601
A. Pérez–Neira, M.A. Lagunas, A. Jové and A. Artés

Viterbi Algorithm with Embedded Channel Estimation Based on Fuzzy Inference 1605
Lorenzo Favalli, Alessandro Mecocci and Pietro Savazzi

Co–channel interference suppression using a fuzzy filter 1609
Sarat Kumar Patra and Bernard Mulgrew

Blind Equalization of Nonlinear Channels Using Hidden Markov Models 1613
Kristina Georgoulakis and Sergios Theodoridis

Symbol–by–Symbol Mobile Radio Channel Equalization Using the K–NN Classifier 1617
Pietro Savazzi, Lorenzo Favalli, Eugenio Costamagna and Alessandro Mecocci

Reduced–complexity Decision Feedback Equalizer for Nonlinear Channels 1621
Francisco J. González–Serrano, Fernando Pérez–Cruz and Antonio Artés–Rodríguez

ThA S–2
Blind Identification / Deconvolution

Source separation without explicit decorrelation 1625
Odile Macchi and Eric Moreau

Blind separation of polarized waves 1629
J.L. Lacoume, F. Glangeaud and J. Mars

Blind Multichannel Estimation Exploiting the Finite Symbol Alphabet 1633
Jaouhar Ayadi and Dirk T.M. Slock

Robustness of the least–squares and the subspace methods for blind channel Identification/equalization with respect to channel undermodeling 1637
Athanasios P. Liavas, Phillip A. Regalia and Jean–Pierre Delmas

Identifiability Conditions for Blind and Semi–Blind Multichannel Estimation 1641
Elisabeth de Carvalho and Dirk T.M. Slock

Blind and Informed Cyclic Array Processing for Cyclostationary Signals 1645
Pascal Chevalier and Alain Maurice

A numerical algorithm for implementing the single–stage maximization criterion for multichannel blind deconvolution 1649
Shuichi Ohno, Yujiro Inouye and Kazuaki Tanebe

DFT Based Optimal Blind Channel Identification 1653
Claudio Becchetti, Giovanni Jacovitti and Gaetano Scarano

Blind Identification of IIR Model Based on Outpul Over–Sampling 1657
Lianming Sun, Wataru Nisizawa, Wenjiang Liu and Akira Sano

Blind Multichannel Equalization with Controlled Delay 1661
A. Touzni and I. Fijalkow

ThA S–3
Image Analysis

A Disjoint set Algorithm for the Watershed Transform 1665
Arnold Meijster and Jos B.T.M. Roerdink

Electrostatic formulation for adaptive dilation 1669
Olivier Lavialle and Pierre Baylou

Textured Regions Extraction Using matching pursuit Method 1673
Takahide Kayanuma and Nozomu Hamada

Hexagonal Wavelet Decomposition for Texture Characterization 1677
Aleksandra Mojsilović, Srdan Marković and Miodrag Popović

Optimized Wold–Like Decomposition of 2D Random Fields 1681
Patrizio Campisi, Giovanni Iacovitti and Alessandro Neri

A New Algorithm CGA for Image Labeling 1685
Guo dong Guo, Shan Yu and Song de Ma

Maximum Entropy Contouring and Clustering for Fractal Attractors with Application to Self–Similarity Coding of Complex Texture 1689
Kohji Kamejima

Region Based Analysis of Video Sequences with a General Merging Algorithm 1693
Luis Garrido and Philippe Salembier

A Framework for Interactive Video Sequence Segmentation Based on Multiple Features 1697
R. Castagno and T. Ebrahimi

Google Exhibit No. 1004
Page 029

Google Exhibit No. 1063
Page 0230

Curvature Estimation of Oriented Patterns 1701
M. Donias, P. Baylou and N. Keskes

ThA SS–4

Multimedia Signal Processing

ARHON: A Multimedia Database Design for Image Documents 1705
K.V. Chandrinos, J. Immerker and P.E. Trahanias

A Multimedia System for the Surveillance of Unattended Railway Stations 1709
Elena Stringa, Claudio Sacchi and Carlo S. Regazzoni

Down–Sampling of Compressed Images in the DCT Domain 1713
A.N. Skodras and C.A. Christopoulos

Broadband Multimedia Systems Architecture and Integration 1717
Irek Defée

A Color Segmentation and Classification Scheme for Facial Image and Video Retrieval 1721
N. Herodotou, K.N. Plataniotis and A.N. Venetsanopoulos

Algorithms for Compressed Video Processing in Multimedia Applications 1725
Francesca Dardi and Giovanni L. Sicuranza

Detecting and Classifying Video Shot Boundaries in MPEG Compressed Sequences 1729
Irena Koprinska and Sergio Carrato

Image Region Extraction for Content–Based Image Retrieval 1733
D. Androutsos, K.N. Plataniotis and A.N. Venetsanopoulos

ThA SS–5

Transform Methods in Signal Processing

Wavelets, filterbanks, and the Karhunen–Loève transform 1737
Michael Unser

Total variation based interpolation 1741
F. Guichard and F. Malgouyres

Transform oriented technology for quantitative analysis of fetal movements in ultrasound image sequences 1745
Leonid Yaroslavsky and Ben–Zion Shaick

Discreete–time linear–phase nearly orthogonal wavelet banks 1749
Tapio Saramäki and Karen Egiazarian

Filtering in Consecutive Fractional Fourier Transform Domains — A Method to Synthesize General Linear Systems for More Efficient Digital and Optical Implementations 1753
M. Fatih Erden, Haldun M. Ozaktas and M. Alper Kutay

Fast Fractional Fourier Transform 1757
E.V. Labunets and V.G. Labunets

Solution of the super–resolution problem through extrapolation of the orthogonal spectra using multi–valued neural technique 1761
Igor Aizenberg, Naum Aizenberg, Jaakko Astola and Karen Egiazarian

New Fast Trigonometric Transforms 1765
K. Egiazarian, V.G. Labunets, E.V. Labunets and J. Astola

Fast Algebraic Convolution for Prime Power Lengths 1769
Reiner Creutzburg and Torsten Minkwitz

ThA P–6a

Statistical and Array Signal Processing

On the issue of rank estimation in subspace tracking: the NA–CSVD solution 1773
Philippe A. Pango and Benoît Champagne

The min–norm beamformer: a new estimate of the propagation speed of waves in a car exhaust 1777
Gema Piñero and Luis Vergara

Direction–of–Arrival Estimation of Cyclostationary Coherent Signals in Array Processing 1781
Jingmin Xin, Hiroyuki Tsuji, Yoshihiro Hase and Akira Sano

A Downlink Adaptive Transmitting Antenna Method for T/F/SDMA FDD Systems Avoiding DOA Estimation 1785
T. Asté, P. Forster, L. Féty and S. Mayrargue

DOA Outlier Mitigation for Generalised Spatial Smoothing 1789
Yuri I. Abramovich and Nicholas K. Spencer

Fast High Resolution Methods 1793
S. Bourennane, M. Frikel and A. Bendjama

Geometrical determination of ambiguities in bearing estimation for sparse linear arrays 1797
Anne Flieller, Pascal Larzabal and Henri Clergeot

Sufficient conditions for the unique localization of distributed sources 1801
Shahrokh Valaee and Benoît Champagne

Spatial and Temporal Processing of Cyclostationary Signals in Array Antennas Based on Linear Prediction Model 1805
Ami Kanazawa, Hiroyuki Tsuj, Jingmin Xin and Blagovest Shishkov

Improving Signal Subspace Estimation and Source Number Detection in the Context of Spatially Correlated Noises 1809
P. Fabry, Ch. Servière and J.L. Lacoume

Blind Separation of Cyclostationary Signals 1813
Adriana Dapena, Daniel Iglesia and Luis Castedo

Blind Seperation from ε–contaminated mixtures 1817
V. Koivunen, P. Pajunen, J. Karhunen and E. Oja

Google Exhibit No. 1004
Page 030

Google Exhibit No. 1063
Page 0231

ThA P–6b

Spectral Estimation and Signal Analysis

Efficient Stochastic Maximum A Posteriori Estimation for Harmonic Signals 1821
Christophe Andrieu and Arnaud Doucet

A Fuzzy Reasoning Based Arma Order Selection Method 1825
Miki Haseyama and Hideo Kitajima

Spectral Methods for Stationary Harmonizable Alpha–Stable Processes 1829
G.A. Tsihrintzis, P. Tsakalides and C.L. Nikias

Robust Bayesian Spectral Analysis via MCMC Sampling 1833
Arnaud Doucet and Christophe Andrieu

Linear Prediction Modeling for Signal Selective DOA Estimation Based on Higher–Order Statistics 1837
Hiroyuki Tsuji, Jingmin Xin, Yoshihiro Hase and Blagovest Shishkov

Wavelet–packet basis selection for abrupt changes detection in multicomponent signals 1841
Eric Hitti and Marie–Françoise Lucas

Detection of cisoids in noise 1845
Joakim Sorelius

High resolution nearly–ML estimation of sinusoids in noise using a fast frequency domain approach 1849
Malcolm D. Macleod

A Period Detector for Pseudo–Periodic Signals 1853
Bart F. Rice

Relationship between the Wigner–Distribution and the Teager Energy 1857
R. Hamila, F.A. Cheikh, J. Vesma, J. Astola and M. Gabbouj

LPC and CCF Vocal Tract Models in Speech Synthesis 1861
Robert Vích and Zdeněk Smékal

Real Time Detectror for Cyclostationary RFI in Radio Astronomy 1865
Rodolphe Weber and Christian Faye

ThA P–7

Filter Design and Subband Techniques

Ladder Scheme for Perfect Reconstruction Modulated Filter Banks 1869
M. Gharbi, M. Colin, M. Gazalet and F.X. Coudoux

Multistage Implementation of Optimal Reconstruction in Noisy Filter Banks 1873
Onoriu Brădenanu and Ulrich Appel

Optimal Subband Analysis Filters Compensating for Quantization and Additive Noise 1877
Anastasios Doulamis, Nikolaos Doulamis and Anastasios Delopoulos

Multidimensional Multirate Filter Without Checkerboard Effects 1881
Yasuhiro Harada, Shogo Muramatsu and Hitoshi Kiya

Inter–Relationships between Different structures for periodic systems 1885
Desmond McLernon

A Design Approach to a Hierarchical Structure Transversal Filter Using GA 1889
S. Nakamura, T. Hoshi, S. Watanuki and M. Yoneyma

Design of Primitive Operator Digital Filters Using Genetic Algorithms 1893
David W. Redmill and David R. Bull

Design of Equiripple Minimum Phase FIR–Filters 1897
A. Groth and H.G. Göckler

Designing Low–Pass Digital Filters with Fewer paremeters 1901
Gang Li

Cluster Filter 1905
Jari Ili–Hietanen, Konsta Koppinen, Katriina Halonen

Least Squares Design of IIR Filters with Arbitrary Magnitude and Phase Responses and specified Stability Margin 1909
Mathias C. Lang

Quadratic FIR – Filter Design as a Generalized Eigenproblem 1913
Jeffrey O. Coleman

On the Optimal Structure of 2–D Digital Filters with L_2–Sensitivity Minimization 1917
Gang Li

Design of FIR filters for images based on properties of human vision 1921
Naoyuki Aikawa and Masamitsu Sato

Lattice Structure of Transparent Layered Optical Systems 1925
Yoshimi Monden and Hideki Hirayama

Complexity constraints – A way toward simpler structuring systems 1929
Pertti Koivisto and Pauli Kuosmanen

An Effective Algorithm for Filtering and Observer Design 1933
Salim Ibrir

Adaptive System Identification using the Normalized Least Mean Fourth Algorithm 737
Azzedine Zerguine and Maamar Bettayeb

Google Exhibit No. 1004
Page 031

Google Exhibit No. 1063
Page 0232

Volume IV

PLENARY TALK

Audio–Visual Signal Processing for Mobile Communications 1937
P. Haavisto

FM SS–1
Applications of Multirate Digital Signal Processing

N–th Band Filter Design 1943
Tapio Saramäki and Markku Renfors

Fractional Rate Decimator and Interpolator Design 1949
Tor A. Ramstad

Subband Coding of Speech and Audio 1953
Charles D. Creusere

Multiresolution Coding of Image and Video Signals 1957
Bernd Girod, Frank Hartung and Uwe Horn

Acoustic Echo Control in Subbands – An Application of Multirate Systems 1961
Gerhard Schmidt

Transmultiplexers–Some new Applications 1965
R.D. Koilpillai, T.Q. Nguyen and K.C. Zangi

Discrete Wavelet Packet Based Multitone Modulation for Transmitting Complex Symbols 1969
T.K. Adhikary and V.U. Reddy

Oversampling A/D and D/A Converters 1973
Gabor C. Temes

FM S–2
Signal Processing for Communications II

Permutation Spreading in Wavelet OFDM Systems 1977
F. Dovis, M. Mondin and F. Daneshgaran

Performance analysis of a Wavelet–Based Generalized Sidelobe Cancellers 1981
Yi Chu, Wen–Hsien Fang and Shun–Hsyung Chang

A Decorrelating Multiuser Receiver for DS/CDMA Systems Using TCM 1985
Kwang Soon Kim, Hong Gil Kim, Iickho Song, So Ryoung Park, Seok Ho Yoon and Hyung–Myung Kim

Bearings–only Target Motion Analysis by estimation of densities 1989
Marc Spigai, Jean–François Grandin

PLL Frequency Synthesizer System Utilizing Multi–Programmable Dividers 1993
Yasuaki Sumi, Kouichi Syoubu, Shigeki Obote, Yutaka Fukui and Yoshio Itoh

DSP–FLL for FSK Demodulation 1997
Shigeki Obote, Yasuaki Sumi, Kouichi Syoubu, Yoshio Itoh and Yutaka Fukui

Some improvements of a rotation invariant autoregressive method.
Application to the neural classification of noisy sonar images 2001
H. Thomas, C. Collet, K. Yao and G. Burel

A new approach to ECF design is proposed based on IRLS algorithm 2005
Andrija Petrović, Aleksa Zejak, Igor Simić, Bojan Zrnić

FM SS–3
Simulation Based Computational Methods for Statistical Signal Processing

On a New Stochastic Version of the EM Algorithm 2009
Colin Campbell and Simon Godsill

Variable Selection by a reversible jump MCMC approach 2013
Petar M. Djurić

A comparison of sample based filters and the extended Kalman filter for the bearings–only tracking problem 2017
Neil Gordon and Michael Pitt

Some Examples of Inverse Problems in Geophysics 2021
Marc Lavielle and David Marquez

Bayesian Decomposition Trees with Application to Signal Denoising 2025
D. Leporini

MCMC Methods for Restoration of Nonlinearly Distorted Autoregressive Signals 2029
Paul T. Troughton and Simon J. Godsill

Multidimensional Optimisation of Harmonic Signals 2033
Paul J. Walmsley, Simon J. Godsill and Peter J.W. Rayner

FM S–4a
Adaptive Systems II

Undermodeled Equalization: Extrema of the Godard/Shalvi–Weinstein Criterion 2037
Phillip A. Regalia and Mamadou Mboup

Blind Single–Channel Interference Rejection Using Godard's Criterion 2041
D. Nussbaum and O. Macchi

Robust Second–Order Blind Equalization of Polyphase Channels 2045
Constantinos Papadias, David Gesbert and Arogyaswami Paulraj

Google Exhibit No. 1004
Page 032

Google Exhibit No. 1063
Page 0233

A Constant Modulus Approach to Multiple Access Interference Rejection 2049
Joaquín Míguez and Luis Castedo

Constrained Normalized Adaptive Filters for CDMA Mobile Communications 2053
J.A. Apolinário Jr., S. Werner, P.S.R. Diniz and T.I. Laakso

FM SS–4b
Blind Source Separation

Direct exploitation of non–Gaussianity as a discriminant 2057
I.J. Clarke

Performance of Blind Discrete Source Separation 2061
O. Grellier and P. Comon

A Residual Bound for the Mixing Matrix in ICA 2065
Lieven De Lathauwer and Joos Vandewalle

Generalization of a Maximum–Likelihood Approach to Blind Source Separation 2069
Vicente Zarzoso and Asoke K. Nandi

De–noising of experimental signals from pyroelectric sensors by a source separation method 2073
R. Huez, D. Nuzillard and A. Billat

FM S–5
Motion / Compression

Hierarchical Mesh–based motion estimation using a differential approach and application to video coding 2077
Patrick Lechat, Michael Ropert and Henri Sanson

Compact representation of motion parameters for object–based video coding 2081
Roger A. Packwood, Michael K. Steliaros and Graham R. Martin

Object oriented coding using 3D motion estimation 2085
G. Calvagno, V. Orsatti, R. Rinaldo and L. Sbaiz

Adaptive orthogonal filter bank trees for encoding of motion–compensated frame differences 2089
Wolfgang Niehsen and Mathias Wien

Architectures for Vector–Tracing based Motion Estimation for MPEG2 type coding for TV and HDTV 2093
M. Gumm, F. Mombers, S. Dogimont, I. Remi and D. Mlynek

Vector Tracing Techniques for Motion Estimation Algorithms in Video Coding 2097
Marco Mattavelli and Giorgio Zoia

A Background Memory Update Scheme for H.263 Video Codec 2101
Kui Zhang and Josef Kittler

Connected operators for sprite creation and layered representation of image sequences 2105
P. Salembier, O. Pujol and L. Garrido

Low–bitrate Video coding with Third Order Geometric Transformations 2109
Cornelis H. Slump, Marcel A.J.A van Veen and Frederik J. de Bruijn

FM P–6
System Identification and Parameter Estimation

Generalized Quadratic Minimization and a Signal processing application 2113
A. Gorokhov and P. Stoica

Frish filtering of noisy signals 2117
Paolo Guidorzi and Roberto Guidorzi

An Algebraic Approach to the Subset Selection Problem 2121
Ahmed Tewfik and Mohammed Nafie

Person Identification via the EEG Using Computational Geometry Algorithms 2125
M. Poulos, M. Rangoussi and E. Kafetzopoulos

A Vector Quantization Schema for Non–Stationary Signal Distributions Based on ML Estimation of Mixture Densities 2129
N. A. Vlassis, K. Blekas, A. Stafylopatis and G. Papakonstantinou

Fourth–order cumulant–based algorithms for non–minimum phase MA system identification 2133
Diego P. Ruiz, Antolino Gallego and Maria C. Carrion

Stochastic system identification for ATM netwrork traffic models: a time domain approach 2137
Katrien De Cock and Bart De Moor

Blind identification of sparse multipath channels using cyclostationary statistics 2141
Karim Abed–Meraim and Yingbo Hua

On the linearly constrained blind multichannel equalization 2145
Santiago Zazo, José Manuel Paez–Borrallo

Evolutionary Multimodel Partitioning Filters for Multivariable Systems 2149
G.N. Beligiannis, K.G. Berketis, D.A. Fotakis and S.D. Likothanassis

Adaptive AR Model Identification Based on the FAEST Filters 2153
S.D. Likothanassis, E.N. Demiris and D.G. Karelis

Markovian approach to random fields nonlinear detection and estimation problems 2157
A.B. Shmelev

Blind and Semi–Blind Maximum Likelihood Techniques for Multiuser Multichannel identification 2161
Elisabeth de Carvalho, Luc Deneire and Dirk T.M. Slock

FM P–7
Image and Shape Analysis

A Robust Method for Reflections Analysis in Color Image Sequences 2165
Andrea Teschioni and Carlo S. Regazzoni

Google Exhibit No. 1004
Page 033

Google Exhibit No. 1063
Page 0234

Locating text in color document images 2169
Erel Ortaçağ, Bülent Sankur and Khalid Sayood

Characterisation of circular objects based on the Hough Transform 2173
Bento Correia, F. Carvalho Rodrigues

How to detect dominant points on 3–D curves 2177
Kazuhide Sugimoto and Fumiaki Tomita

Curvature Variation of Projected Cross–Sections from Straight Uniform Generalized Cylinders 2181
William Puech and Jean–Marc Chassery

3–D measurements using conoscopy and application to ophthalmology 2185
Christophe Moser, George Barbastathis and Demetri Psaltis

Projection–Based Registration of Radiological Images 2189
Çiğdem Eroğlu and A. Enis Çetin

Lip Features for Speech and speaker Recognition 2193
Ronald Auckenthaler, Jason Brand and John S. Mason

Precise Eye Detection from Image Sequences 2197
Weimin Huang, Jian–Kang Wu, Qibin Sun and Chain–Prong Lam

A New Stereo Matching algorithm Based on Probabilistic Diffusion 2201
Sang Hwa Lee, Jong Il Park and Choong Woong Lee

An efficient balanced hierarchical data structure for multiversion accesses to spatio–temporal data 2205
Hiroyuki Dekihara and Yasuaki Nakamura

Vision Navigation of an Autonomous Vehicle by Fuzzy Reasoning 2209
W. Li and F.M. Wahl

Measuring Tree–Ring Parameters using the Generalised Fisher Ratio 2213
S. Zheng and C.G. Molina

FA S–1
Signal Processing for Communications III

ABRLS Algorithm for Time Variant Channel Equalisation 2217
Tetsuya Shimamura and Colin F.N. Cowan

The Use of Evolutionary Optimisation in Channel Equalisation 2221
F. Sweeney, P. Power and C.F.N. Cowan

An Algorithm for Channel Equalization with Adaptive Tap Position Control 2225
M.T. Arvind and P.G. Poonacha

A Constant Modulus Array for Real Signals 2229
W. Pora, S. Lambotharan, J.A. Chambers and A.G. Constantinides

Blind Iterative Separation of CDMA Signals by Nonlinear Independent Component Analysis 2233
Ari Hottinen

A bayesian method for GPS signals delay estimation 2237
J. Soubielle, I. Fijalkow, P. Duvaut, J.Y. Delabbaye and A. Bibaut

Radio Frequency Interference Rejection in Radio Astronomy Receivers 2241
Peter Fridman

Spatial–Temporal Processing with Restriction of Degrees of Freedom for CDMA–Multiuser Detection 2245
O. Muñoz, J.A. Fernández–Rubio

An Efficient Non Linear Receiver for High Density Optical Recording 2249
L. Agarossi, S. Bellini, F. Bregoli and P. Migliorati

FA S–2a
Image Filtering

Noise Reduction of Image Sequences as Preprocessing for MPEG2 Encoding 2253
P.M.B van Roosmalen, A.C. Kokaram and J. Biemond

Variational Principles Applied to Image Filtering 2257
Lorenzo José Tardón–García, Javier Portillo–García

Removal of mixed noise by adaptive Linear combination of weighted order statistics (LWOS) filters based on local statistics 2261
Akira Taguchi and Tadasuke Inoue

A New Operator for Image Processing Based on Lp–Filter Approximation 2265
M. Tabiza and Ph. Bolon

Signal and Image Denoising in Transform Domain and Wavelet Shrinkage: A Comparative Study 2269
Ruşen Öktem, Leonid Yarovslavsky and Karen Egiazarian

FA S–2b
Watermarking

Generation of the Signature with the Structured Information of the Image 2273
Hirotsugu Kinoshita and Masafumi Satoh

Self–similarity based image watermarking 2277
Patrick Bas, Jean–Marc Chassery and Franck Davoine

Removing Spatial Spread Spectrum Watermarks by non–linear Filtering 2281
Gerrit C. Langelaar, Reginald L. Lagendijk and Jan Biemond

Robust Image Watermarking in the Subband or Discrete Cosine Transform Domain 2285
Dimitrios Tzovaras, Nikitas Karagiannis and Michael G. Strintzis

Non–Invertible Statistical Wavelet Watermarking 2289
Gianluca Nicchiotti and Ennio Ottaviani

Google Exhibit No. 1004
Page 034

Google Exhibit No. 1063
Page 0235

FA S–3
Image and Video Coding

User Interaction in Content–Based Video Coding and Indexing 2293
Paulo Correia and Fernando Pereira

An MPEG2 Compliant PSNR controller for a Constant High Quality Studio Environment 2297
Enrico Frumento and Rosa Lancini

Adaptive Tree–Structured Lattice Vector Quantization for Video Coding 2301
Vincent Ricordel and Moncef Gabbouj

Linear Combination of Face Views for Low Bit Rate Face Video Compression 2305
Ioannis Koufakis and Bernard Buxton

Computational Graceful Degradation for Video Decoding 2309
Marco Mattavelli and Sylvain Brunetton

Cost–Based Region Growing for Fractal Image Compression 2313
Hannes Hartenstein and Dietmar Saupe

Context–based adaptive arithmetic coding for lossy plus lossless image compression 2317
Guang Deng

High Compression of Chrominance Exploiting its Correlation with Luminance 2321
Maciej Bartkowiak and Marek Domański

Progressive Medical Image Compression Using a Diagnostic Quality Measure on Regions–of–Interest 2325
Alberto Signoroni and Riccardo Leonardi

Multidimensional companding for Lattice Vector Quantization of Circularly Symmetric Densities 2329
Stephan F. Simon and Werner Praefcke

FA S–4
Audio and Electroacoustics II

Stochastic gradient algorithms in active control 2333
Alberto González

Stereophonic Acoustic Echo Canceller using single adaptive filter per channel 2337
Eun Sook Kim, Kyu Hwa Jeong, Won Cheol Lee and Dae Hee Youn

Modified Multidelay Adaptive Filter for Acoustic Echo Cancellation 2341
Tõnu Trump

Audio Subband Coding with Improved Representation of Transient Signal Segments 2345
Jörg Kliewer and Alfred Mertins

Adaptive context based sequential prediction for lossless audio compression 2349
Ciprian Doru Giurcăneanu, Ioan Tăbuş and Jaakko Astola

Active Binaural Sound Localization 2353
Greg Reid and Evangelos Milios

Recognition of Isolated Musical Patterns using Discrete Observation Hidden Markov Models 2357
Aggelos Pikrakis, Sergios Theodoridis and Dimitris Kamarotos

Toward the automatic synthesis of nonlinear wave digital models for musical acoustics 2361
Federico Pedersini, Augusto Sarti, Stefano Tubaro and Roberto Zattoni

Number Theoretical Means of Resolving a Mixture of Several Harmonic Sounds 2365
Anssi Klapuri

FA S–5a
System Identification

Identification of linear time–variant systems by spectral correlation measurements 2369
Luciano Izzo and Antonio Napolitano

A Polynomial–Algebraic Method for Non–Stationary TARMA Signal Analysis 2373
R. Ben Mrad, S.D. Fassois and J.A. Levitt

Performance analysis of some methods for identifying continuous–time autoregressive processes 2377
Torsten Söderström and Magnus Mossberg

The bicepstral distance between random signals: a new tool for comparison of arma models identification methods based on higher–order statistics 2381
Jean–Luc Vuattoux and Eric Le Carpentier

Discrete Time Equivalents of Linear Systems Driven by White Noise 2385
N. Martins and A.C. Rosa

FA S–5b
Higher Order Statistics

An HOS based Statistical Test for Multiplicative Noise Detection 2389
Martial Coulon and Jean–Yves Tourneret

Bussgang Test: A Powerful Non–Gaussianity Test 2393
Caetano Giunta, Giovanni Jacovitti and Gaetano Scarano

Nonlinear Constrained Optimization Using Lagrangian Approach for Blind Source Separation 2397
Benoit Stoll and Eric Moreau

Non–Minimum Phase AR Identification Using Blind Deconvolution Methods 2401
Gaetano Scarano and Giampiero Panci

Limitations of the ARMA w–Slice Method Using a Linear Combination of Higher–Order Cumulants 2405
Damjan Zazula and Jean–Luc Vuattoux

Modulation Classification Based on a Maximum–Likelihood Receiver in the Cyclic–Hos Domain 2409
Pierre Marchand, Christophe Le Martret and Jean–Louis Lacoume

Google Exhibit No. 1004
Page 035

Google Exhibit No. 1063
Page 0236

FA P–6

Applications

Localization Error Analysis of a Fast, Broadband Matched Field Method for a Single Hydrophone 2413
Erland Sangfelt and Bernt Nilsson

Cascaded Scattering Functions for Sonar Signal Processing 2417
Lora G. Weiss and Leon H. Sibul

A Fast Near–Field Bearing Estimation Via Spatial Exponential Kernel Distribution 2421
S.–H. Chang, W.–W. Lin and W.–H. Fang

A system for Seismic Data Processing 2425
Klaus Köster and Michael Spann

A Feature Extractor of Seismic Data Using Genetic Algorithms 2429
A.V. Adamopoulos, S.D. Likothanassis and E.F. Georgopoulos

Classification Microstructure of Human Sleep Using EEG Modelling 2433
José M. Allen Lima and Agostinho C. da Rosa

Wavelet based neural network architecture for ECG signal compression 2437
Shubha Kadambe and Pramila Srinivasan

Process–adapted Biosignal Processing – an Approach of Wavelet Transformation to Intracardial Leads 2441
R. Poll, T. Rauwolf and E. Meisel

Wavelet–Entropy Applied to Brain Signal Analysis 2445
O.A. Rosso, R. Quian Quiroga, S. Blanco, A. Figliola and E. Başar

Denoising of ECG signals using Wavelet Shrinkage with time–frequency dependant treshold 2449
Nikolay Nikolaev and Atanas Gotchev

Hidden Markov models compared to the wavelet transform for P–wave segmentation in EGC signals 2453
L. Clavier, J.M. Boucher and E. Polard

Hierarchical Neural Network System to Breasts Cancer Diagnosis 2457
L.F. Mingo, J. Castellanos, V. Giménez and A. Vilarrasa

An Instantaneous Frequency Dispersion Estimator for Detecting High Intensity Transient Signals in Human Blood Flow 2461
Emmanuel Roy, Silvio Montrésor, Pierre Abraham and Marc Baudry

Detection of microcalcifications using non–linear filtering 2465
Dirk Meersman, Paul Scheunders and Dirk Van Dyck

ECG compression using PCA with polynomial estimation of higher order coefficients 2469
David Hamilton, William Sandham and Alberto Blanco

Complete Coding Scheme Using Optimal Time Domain ECG Compression Methods 2473
Ranveig Nygaard and Dag Haugland

Segmentation with Predictive Error and Recursive Likelihood Ratio Deviation methods in material characterization 2477
S. Femmam and N.K. M'Sirdi

Adaptive Directional Order Filters and Mathematical Morphology for Road Network Extraction on SAR Images 2481
Jocelyn Chanussot, Imad Issa and Patrick Lambert

Fractal Estimation in a Given Frequency Range. Application to Smectite Images 2485
R. Harba, M. Cintract, M. Zabat and H. Van Damme

Radio Frequency interferences suppression for ultra Wide Band radar 2489
Bruno Juhel, Georges Vezzosi and Marc Le Goff

FA P–7

Image Analysis

Segmentation of Urban Areas in Spot images using MRF 2493
F. Richard, F. Falzon, J. Zerubia, and G. Giraudon

A supervised Lloyd algorithm and segmentation of handwritten Japanese characters 2497
Fujiki Morii

Image decomposition capabilities of the joint wavelet and radon transform 2501
G. Olmo, L. Lo Presti and E. Magli

Segmentation of Boron Carbide Microscopic Images which present twins 2505
Wanessa Nascimento Matta, Arnaldo de Albuquerque Araújo, Marcos Carneiro de Andrade and Gilles Bertrand

Information Criteria for Histogram Thresholding Techniques 2509
Pierre Courtellemont, Christian Olivier and Frédéric Jouzel

CT image labeling using simulated annealing algorithm 2513
Zoran Majcenić and Sven Lončarić

Image Segmentation by a Multiresolution Approach 2517
Nacera Benamrane, Karima Kies and Jun Shen

Invariant Supervised Texture Recognition Using Multi–Channel Gabor Filters 2521
Manfred Bresch

Supervised texture classification – Selection of moment lags 2525
Vladimiros Antoniades and Asoke K. Nandi

Unsupervised texture segmentation using discrete wavelet frames 2529
S. Liapis, N. Alvertos and G. Tziritas

Scale Invariant Texture Classification with Mathematical Morphology 2533
Virginia L. Ballarin, Emilce G. Moler and Marcel Brun

An Identification method for Texture AR–2D Modelling Based on Auto– and Partial Correlation Measures 2537
Isabelle Claude and André Smolarz

Google Exhibit No. 1004
Page 036

Google Exhibit No. 1063
Page 0237

Mean Depth–Width Ratio of Extrema as Textural Feature
for Automated Cell Proliferation Analysis 2541
Corkidi Gabriel, Vega Leticia and Márquez Jorge

On the Initial Label Configuration of MRF 2545
Guo dong Guo, Shan Yu and Song de Ma

Curvature Scale Space Based Image Corner Detection 2549
Farzin Mokhtarian and Riku Suomela

Road boundaries detection using Color Saturation 2553
Pierre Charbonnier, Philippe Nicolle, Yannick Guillard and Jean Charrier

Directional second order derivatives: application to edge and corner detection 2557
S. Guillon and M. Donias

Linear feature detectors and their application to cereal inspection 2561
E.R. Davies, D.R. Mason, M. Bateman, J. Chambers and C. Ridgway

A Probabilistic Framework for the Hough Transform and Least Squares Pose Estimation 2565
Jorge S. Marques

Proceedings

Google Exhibit No. 1004
Page 037

Google Exhibit No. 1063
Page 0238



Figure 4: Actual and reconstructed frame from the "Miss America" test sequence

control points), and for 30 frames/sec the compressed video stream requires 47.7 Kbits/sec and 41.6 Kbits/sec respectively. Figure 4 shows an actual and a reconstructed frame of the "Miss America" video sequence[1]. This face was synthesised by 19 control points and the 10 most important principal components, quantized at 12 bits, for each eye and the mouth. These were computed from a training set of 30 images extracted from the first 30 frames of the sequence.

5 DISCUSSION AND CONCLUSIONS

In this paper, we propose a simple encoding technique that can be used for the compression of face video data for low bit-rate videoconferencing applications. By using knowledge that the video data contains face images against a fairly uniform and stationary background, the proposed scheme encodes the face and background area in each frame as a linear combination of three basis views. Important facial features such as the eyes and mouth that change a lot during conversation or with changes of expression,may be encoded using principal components analysis. As a subset of the control points used in the linear combination of views are used in the PCA, features reconstructed by PCA may easily be merged into the face reconstructed from the basis views.

The linear combination method can achieve very high compression ratios since for every new frame it does not transmit any textural information; only shape information needs to be transmitted. Also, it is computationally simple since only implicit $3D$ information derived from the three basis $2D$ face views is used. Furthermore, it preserves the temporal and spatial resolution of the video data. However, sparse correspondence between the basis and novel face views is needed, so as in other knowledge-based coding approaches, a robust and fast scheme for detecting faces and control points, along with elimination of control points outliers, is required. The quality of the reconstructed face images can be further improved by eliminating self-occlusion effects by hidden surface removal [10].

[1]Grey-scale versions of the frame are shown here, although our experiments were performed with colour images

References

[1] M. Liou. Overview of the px64kbits/s Video Coding Standards. *Communications of the ACM*, 34(4):59–63, April 1991.

[2] D.E. Pearson. Developments in Model-Based Video Coding. *Proceedings of the IEEE*, 83(6):893–905, June 1995.

[3] K. Aizawa and T.S. Huang. Model-Based Image Coding: Advanced Video Coding Techniques for Very Low Bit Rate Applications. *Proceedings of the IEEE*, 83(2):259–271, February 1995.

[4] S. Ullman and R. Basri. Recognition by Linear Combination of Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, October 1991.

[5] A. Shashua. *Photometry and Geometry in 3D Visual Recognition*. PhD thesis, Massachusetts Institute of Technology, 1992.

[6] G.D. Hager and M. Toyama. Xvision: A General Purpose Substrate for Real-Time Vision Applications. To appear in Computer Vision and Image Understanding.

[7] A. Goshtasby. Piecewise Linear Mapping Functions for Image Registration. *Pattern Recognition*, 19(6):459–466, 1986.

[8] S. Avidan and A. Shashua. Novel View Synthesis in Tensor Space. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1034–1040, 1997.

[9] D. Shah, S. Marshall, and W.J. Welsh. Principal Components Analysis of Mouth Sequences. *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 5:373–376, 1993.

[10] S. Ullman. *High-Level Vision: Object Recognition and Visual Cognition*, chapter 5. Recognition by the Combination of Views, pages 95–157. MIT Press, 1996.

COMPUTATIONAL GRACEFUL DEGRADATION FOR VIDEO DECODING

Marco Mattavelli, Sylvain Brunetton
Integrated Systems Center - C3i,
Swiss Federal Institute of Technology,
Lausanne CH-1015, SWITZERLAND
Tel: +41 21 6936979; fax: +41 21 6934663
e-mail: Marco.Mattavelli@epfl.ch

ABSTRACT

The implementation of software based video decoders is an advantageous solution over traditional dedicated hardware real-time systems. However, guaranteeing real-time performance, needed when processing video/audio bit-streams, by scheduling processing tasks of variable load and that might exceed the available resources, is an extremely difficult and in many cases an impossible challenge. In this paper we introduce the motivations and ideas of the Computational Graceful Degradation (CGD) approach for the implementation of video decoding. The goal is to provide results of possibly lower visual quality, but respecting the real-time constraints, when the required processing needs exceed the available processing resources. This approach is very attractive since it enables a more efficient usage of the hardware processing power without needing to implement resources for worst case complexity decoding. We show how such technique can be applied to the decoding of compressed video sequences. Simulation results for MPEG-4 and H.263 video compression standards showing some of the interesting achievements and potentialities of CGD approach are also reported. The presented approach can be virtually applied to any video compression standard and processor based platform, although the best performances are achieved when complexity prediction information provided as side information by the video standard is available. This option has been included in the new MPEG-4 standard.

1 INTRODUCTION

The complexity of video decoding is a fluctuating function of time [1]. By complexity we mean the processing time necessary to decode a compressed video bitstream. Even considering a specific processor platform, a specific compression standard, with a relative level and profile, the complexity of video decoding is a variable quantity. Dedicated hardware platforms can guarantee worst case complexity decoding. In contrast, in the case of programmable platforms such a guarantee would imply the allocation of a processing power up to two orders of magnitude higher than the one actually needed for an average decoding [1]. Conversely, a theoretical average value is not useful for an efficient scheduling of a decoding task. The average value for each considered sequence is not stationary, and its variation around the theoretical average one are so unpredictable that no useful scheduling policy can be realized. The first straightforward option is a sustained over-sizing of the processing resources. The second is to accept interruptions in the audio/video output streams when the needed decoding resources exceed the ones available in the system. Obviously both options are not desirable being respectively not economical, or providing a quality of the services which may easily result perceptually unacceptable. The main rationale of the Computational Graceful Degradation (CGD) approach is to provide results of poorer quality when the results of the desirable quality cannot be obtained in real time. When the complexity of the task exceeds the available processing power, real-time services, possibly of degraded quality, are nevertheless guaranteed respecting all real-time constraints. This approach enables the correct scheduling of real-time tasks even when the computational system load fluctuates above the system capabilities.

The concept of computational graceful system degradation using imprecise results has previously been introduced in the field of real-time computing with applications to control system and automation. In these contexts imprecise results are defined as intermediate results with the property that the more recent and intermediate result is the closer one that approximates the final result (i.e. the idea only works in monotonically improving computations) [2, 3, 4]. In these cases the problems to be solved are more concerned with fault tolerance, I/O overloads, stability and reactivity to unexpected inputs, and the corresponding techniques are not suitable for the purposes of video decoding. In our case, the goal is to guarantee real-time video decoding with possible degradations of the image quality, i.e reductions of the peak signal to noise ratio (PSNR), without loosing parts of the image content. Such al-

Google Exhibit No. 1004
Page 038

Google Exhibit No. 1063
Page 0239

ternative is certainly much more desirable than other straightforward options such as cutting image portions or simply skipping frames. These latter options, as can be easily understood, can have catastrophic consequences on prediction based coding systems. In H.263 decoders, for instance, skipping the decoding of a single frame or a portion of it, implies the skipping of all future incoming frames (or portion of them) before the next incoming Intra frame. The time interval of such skipping operation might cover many seconds and the information content loss might be very relevant. Such abrupt interruptions, which are perceptually very annoying, might be caused not only by the simple fluctuation of the decoding complexity load, but also by the fluctuations of the load of other non-video processing tasks which reduce the available processing resources. Moreover, such events in general are out of the control of the video decoder system and cannot be easily forecasted. Considering these facts, it is evident the importance of techniques able to filter out complexity decoding peak loads or able to reduce the average decoding computation load without causing perceptually annoying consequences. Skipping B-VOP could be, in some cases, a viable option accepting the introduction of motion judder in the sequence reproduction. Unfortunately the presence of B-frames or B-VOP cannot be guaranteed in any bit-stream and just at the right time for which a reduction of the complexity is needed to respect the real time constraints.

In contrast with what could be commonly thought, also the classical multi-resolution coding features such as spatial or temporal scalabilities are not really useful for our goals. In general, they are used to optimize the transmission bandwidth usage enabling the compatibility of services at different titrates and spatial resolutions, but they must be embedded in the original bit-stream at the encoder side. If they are not embedded, the changes of resolution at the decoder size can results more computationally expansive than the simple decoding at the given resolution.

From these arguments the usefulness of CGD and its important features and applications are clear. The last key point that has left CGD as a good idea, but of difficult introduction in video coding, was that it was not clear how to implement it efficiently. Now that a technique able to predict with a good degree of approximation the required decoding time of each decoding process is demonstrated and available for MPEG-4 systems [5], we have the key for the useful and efficient implementation of CGD techniques.

2 PRINCIPLES OF DECODING USING THE CGD APPROACH

The decoding operations of a video compression algorithm can be divided into two main categories: 1) operations that cannot be simplified since they may prevent

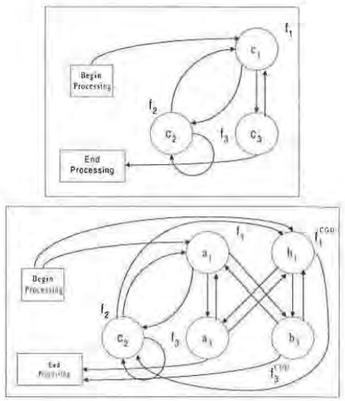


Figure 1: Upper diagram, example of a complexity model representing the data dependent execution of three algorithms. Lower diagram, example of the same complexity model of the upper diagram for which two algorithms (f_1 and f_3 are associated to new CGD algorithms (f_1^{CGD} and f_1^{CGD}).

the extraction and usage of relevant coding information, 2) operations that can be simplified, possibly leading to imprecise or degraded video results. To the first category typically belong all bit-stream parsing operations, entropy decoding (i.e. VLD and arithmetic coding), and all operations related to synchronization and to the syntax of the decoding process. For these operations no degradation mode definition is possible, otherwise it would imply skipping image portions or entire frames since either the data or the coding modes are lost. To the second category belong operations such as the IDCT, the prediction with and without motion compensation in its various forms (PB/B-frame prediction, overlapped block motion compensation (OBMC), half pixel interpolation, macroblock padding and so on). Simplifications of such operations applied to pixels or transform coefficients are always possible but at the expenses of imprecise results.

Two are the possible strategies for the implementation of CGD features. The first is to monitor the execution of the decoding process, to assume that the decoding time of each portion of the frame or VOP is proportional to the image area, and to execute CGD algorithms whenever the current monitored decoding time exceeds the current target decoding time suitable to respect the specified real-time constraints. Such simple strategy has several drawbacks. One is that there is no guaran-

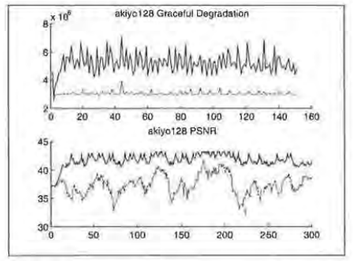


Figure 2: Example of CGD decoding of 300 frames of the sequence Akiyo at compressed at 128 kb/sec with H.263. The upper diagram reports the number of processor cycles for the original decoding (upper curve) and for the CGD decoding lower curve, versus the frame number. The lower diagram reports the two PSNR.

tee that the target decoding time is achieved. Another problem is due to execution of CGD algorithms that *a posteriori* could have not been necessary, thus introducing unnecessary image degradations. Moreover, such degradations are concentrated on critical image areas, and due to the drift given by the degraded reference for the prediction based algorithms, such degradations becomes visible and might seriously affect the resulting sequence PSNR. An example of such approach is reported in Fig. 2. The H.263 decoder has been modified by implementing computationally degraded IDCT and motion compensation algorithms. The interesting result achieved is that the decoding is obtained at a constant processing time for each frame. Unfortunately the PSNR degradations although limited to some dBs are nevertheless relevant from the perceptual point of view.

The second strategy is to exploit the complexity information carried by the frames or VOP headers, like it can be done for MPEG-4 systems [6], and use it in two stages. The first stage predicts the decoding time needed for the current frame or VOP, as described in [5]. The second, on the basis of the predicted decoding time and on the complexity information (i.e. statistics of the algorithms necessary to correctly decode the frame), executes only the CGD algorithms that are just necessary to not exceed the target decoding time.

More formally, let us consider a complexity model similar to the one illustrated in Fig. 1 composed by a set of I algorithms f_i, with $i = 1 \dots I$, and suppose to be able to design for all, or only for a subset S^{CGD} of the functions f_i composing the model, new functions f_i^{CGD} which have the following three properties: 1. the execution of a function f_i^{CGD} yields results that are an approximation of the ones produced by the original function f_i, 2. the execution of f_i^{CGD} does not prevent the execution of any subsequent decoding functions, 3. the weight w_i^{CGD} associated to f_i^{CGD} that represent the execution time of the f_i^{CGD} algorithm for the considered decoder implementation is lower than the corresponding w_i associated to f_i (i.e. $w_i^{CGD} < w_i$). A schematic representation of a complexity model which includes new degraded functions is reported in Fig. 1. The picture shows that new degraded algorithms can be executed at the place of the original algorithms. Obviously, the execution of any f_i^{CGD} may produce PSNR degradations in the reconstruction of some image parts, it does not prevent from correctly decoding the rest of the frame or VOP, but it reduces the necessary decoding time. Let us partition the functions constituting the complexity model into two disjoint sets: S^{Orig} and S^{CGD} respectively for the ones that do not admit a CGD version (f_2 for instance in Fig. 1) and for the ones for which it is possible to design a CGD version (f_1 and f_3). Let us consider for simplicity an image sequence composed by a sequence of single frames or VOPs and suppose that the execution time for the decoding of the frame or VOP is larger than the allocated decoding time:

$$T(t) > T_{Alloc}, \tag{1}$$

where T(t) is the decoding time of the VOP at hand at the sampling time t. Since the decoding time can be approximated by its prediction [5] it can be expressed by:

$$T(t) \simeq T_{Pred}(t) = \sum_{i=1}^{I} c_i w_i, \tag{2}$$

where the c_i are the actual occurrences of algorithm f_i in the current frame or VOP [5], and the w_i are the corresponding complexity weights. The lower limit for the decoding time executing all CGD algorithms f_i^{CGD} at the place of f_i for all $i \in S^{CGD}$, is given by:

$$T_{CGD} = \sum_{i \in S_{CGD}} c_i w_i^{CGD} + \sum_{i \in S_{Orig}} c_i w_i. \tag{3}$$

If $T_{CGD} < T_{Alloc} < T_{Pred}$, is possible to guarantee the real-time constraints by executing the f_i^{CGD} algorithms and accepting approximate decoding results. It is also possible to find for each c_i with $i \in S^{CGD}$, a couple of coefficients a_i, b_i, for which $a_i + b_i = c_i$, and denoting as $T_{Process}$ any decoding time interval which respects the real time constraints, we can express the desired process decoding time as any decoding time value that satisfy the expression:

$$T_{process} = \sum_{i \in S_{CGD}} (a_i w_i^{CGD} + b_i w_i) + \sum_{i \in S_{Orig}} c_i w_i \leq T_{Alloc}. \tag{4}$$

Obviously for any Allocated Decoding Time the goal is to respect the real-time constraints with the lowest

Google Exhibit No. 1004
Page 039

Google Exhibit No. 1063
Page 0240

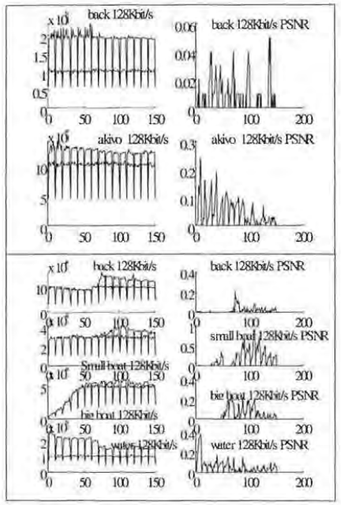


Figure 3: Left, decoding time results for the various VOP of sequences Akiyo and Coastguard, with and without imposing a bound. Right, the PSNR degradations versus the sequence VOP number.

degradation in the video/audio content. Associating a degradation measure to each w_i^{CGD} it is possible to minimize the degradation for any given T_{Alloc}. The challenge is to find the set $S_{a,b}$ of couple of coefficients a_i, b_i, with $i \in S_{CGD}$, that satisfy minimize the video degradations expressed as difference with the non degraded output:

$$S_{a,b} = \text{argmin(a, b)}_{i \in S_{CGD}} : (\text{PSNR} - \text{PSNR(a, b)}). \quad (5)$$

The determination of the set $S_{a,b}$ can be done by means of many suitable optimization method or heuristics. The described techniques can be easily extended to the case of sequences containing multiple VOP at the same sampling time t.

3 SIMULATION RESULTS

Simulation results demonstrate the possibility of successful implementations of CGD for MPEG-4 video decoding. A bound on the decoding complexity of each VOP is defined setting a maximum constant T_{Alloc}. Figure 3 report the results of MPEG-4 video decoding imposing such bounds on the decoding time of each video object plane (VOP) for the test images Coastguard and Akiyo coded at 128 kb/s. On the left, the graphs report the decoding time in msec for the unbounded and bounded decoding (lower curve) versus the image sequence number. On the right the graphs reports the resulting PSNR degradation. The decoder is the Momusys CD on a Sun Ultrasparc station modified implementing degraded motion compensation algorithms. Obviously the actual results depend on the decoder, on the platform on the CGD algorithms implemented, but the good results demonstrate that bounded complexity decodings are possible with minimal image degradations much lower than the one obtained without complexity information and decoding time prediction.

4 CONCLUSION

We have introduced the motivations and concepts of the CGD approach for the implementation of video decoders on processor based platforms. Simulation results for MPEG-4 video decoding demonstrate the successful implementation of the CGD approach. Obviously, the implementation of CGD features are not proposed to be used at their maximum degradation rate but as tool able to guarantee real-time performance in presence of peaks of decoding complexity or as a way to ease the scheduling problems given by temporary fluctuations of the available processing power. Moreover, the presented approach may enable the successful implementation of video decoding for levels and profiles that would be forbidden if the complexity of *worst case decoding* has to be guaranteed.

References

[1] M. Mattavelli and S. Brunetton. Scalability of the mpeg-4 vm based on the computational power of texture decoding. Technical Report Document M925, ISO/IEC JTC1/SC29/WG11, Tampere, Finland, July 1996.

[2] A.D. Stoyenko W.A. Halang. *Constructing predictable real time systems.* Kluwer Academic Publisher, 1991.

[3] J. Liu K. Lin, S. Natarjan. Imprecise results: Utilizing partial computations in real-time systems. In *Proc. of IEEE Real-Time System Symp.*, Dec. 1987.

[4] S. Natarjan J. Liu, K. Lin. Scheduling real time periodic jobs using imprecise results. In *Proc. of IEEE Real-Time System Symp.*, Dec. 1987.

[5] M. Mattavelli and S. Brunetton. Prediction of decoding time for mpeg-4 video. In *Proc. of Eusipco 98, IX European Signal Processing Conference*, Island of Rhodes, Greece, September 8-11 1998.

[6] ISO-IEC/JTC1. Doc for iso/iec cd 14496-2 visual. Technical Report Doc. W2119, Tokyo, March 1998.

COST-BASED REGION GROWING FOR FRACTAL IMAGE COMPRESSION

Hannes Hartenstein, Dietmar Saupe
Institut für Informatik, Universität Freiburg,
Am Flughafen 17, 79110 Freiburg, Germany
Tel: +49 761 203 8185; fax: +49 761 203 8182
e-mail: hartenst,saupe@informatik.uni-freiburg.de

ABSTRACT

For application in fractal coding we investigate image partitionings that are derived by a merge process starting with a uniform partition. At each merging step one would like to opt for the rate-distortion optimal choice. Unfortunately, this is computationally infeasible when efficient coders for the partition information are employed. Therefore, one has to use a model for estimating the coding costs. We discuss merging criteria that depend on variance or collage error and on the Euclidean length of the partition boundaries. Preliminary tests indicate that improved coding costs estimators may be of crucial importance for the success of our approach.

1 INTRODUCTION

In fractal image compression the image to be coded is partitioned into image blocks called *ranges*. Each range is approximated by another image part, called *domain*, which is scaled geometrically and modified using an affine transformation for the grey values. The fractal code consists of the partitioning information and, for each range, the domain block indices and the transformation parameters. Under certain constraints this code represents a contractive transformation whose unique fixed point is an approximation to the original image.

One of the bit allocation problems of fractal coding is to decide how many bits should be spent for the partitioning information. The most widely used strategies to partition the image are hierarchical methods like quadtree or horizontal/vertical partitioning. Here the coding costs for the partition are relatively low. On the other hand, more complex partitionings could lead to lower approximation errors. A region-growing based approach was introduced to fractal compression in [1]. This approach was extended in [2, 3] leading to a fast fractal coder with coding results comparable to Fisher's HV-coder [4].

Image partitioning problems can be stated as optimization problems where a specific cost function has to be minimized [5]. In this paper we present a formulation of the partitioning problem in fractal coding along these lines and analyze two cost functions that include the segmentation coding costs. The study was motivated by the question whether adding a segmentation cost constraint to the cost function will improve coding results, and by the need to determine properties of *good* partitionings.

This paper is organized as follows: In Section 2 we give basic definitions and state the partitioning problem as an optimization problem. We also comment on the inherent complexity of this problem. In Section 3 we outline the partition coding scheme used in this study. In Sections 4 and 5 algorithms for region-based fractal coding are presented. We report on coding results in Section 6 and give examples of decoded images.

2 OPTIMIZATION PROBLEM

Let $I : S = \{0, ..., 2^n - 1\} \times \{0, ..., 2^n - 1\} \to \{0, ..., 255\}$ be an image, and let $\hat{I}$ be the image of size $2^{n-1} \times 2^{n-1}$ that is obtained from I via pixel averaging, i.e., $\hat{I}(i, j) = \sum_{0 \le k,m \le 1} I(2i + k, 2j + m)$.

A partition $\mathcal{P}$ for I is a set of subsets of S such that $\bigcup_{P \in \mathcal{P}} = S$, $P \cap Q = \emptyset$ for $P, Q \in \mathcal{P}, P \neq Q$, and each $P \in \mathcal{P}$ is connected in the 4-neighbor sense. I restricted to a region $P \in \mathcal{P}$ is called a range of I.

Given a partition $\mathcal{P}$, the fractal code for image I is determined as follows. For each range R of this partition a block D_R of $\hat{I}$ (or of $\hat{I}$ under a square isometry) and real parameters s_R, o_R are sought such that

$$\|R - s_R D_R - o_R \mathbf{1}\|^2 = \min_D \min_{|s|<1, o \in \mathbb{R}} \|R - s^q D - o^q \mathbf{1}\|^2.$$

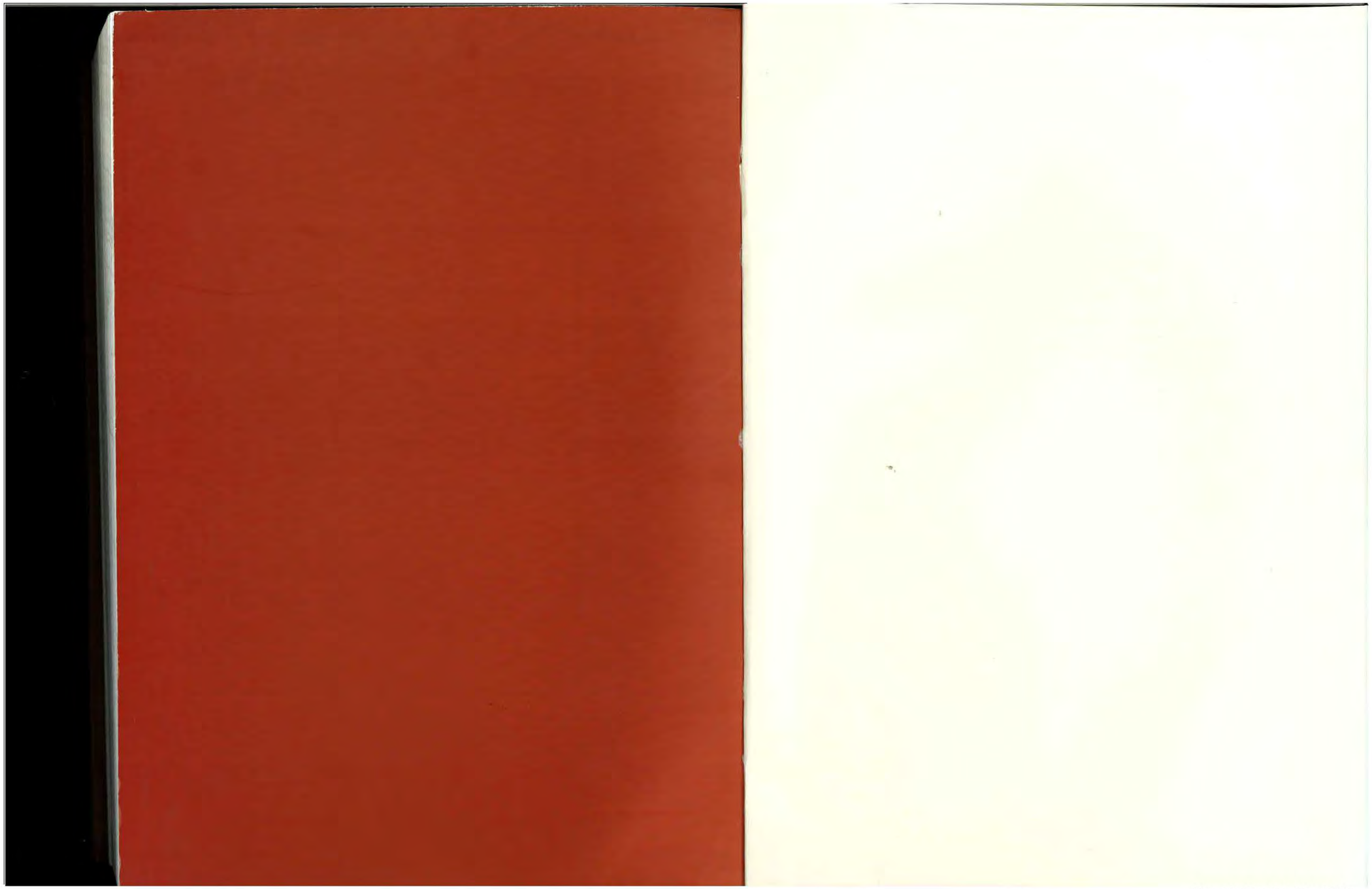
Here, $\|\cdot\|$ stands for the Euclidean norm, $\mathbf{1}$ is a vector with a 1 in each component, and s^q, o^q stand for the quantized *luminance* parameters. If R corresponds to region P we set $E(P) := \|R - s_R D_R - o_R \mathbf{1}\|^2$, and the overall approximation error, called *collage error*, is defined by

$$E(\mathcal{P}) := \sum_{P \in \mathcal{P}} E(P).$$

The global optimization problem is to find in the space of all partitions the optimal one in the sense that the rate of the corresponding fractal code is lower than a given budget limitation while simultaneously the overall

Google Exhibit No. 1004
Page 040

Google Exhibit No. 1063
Page 0241



Ex. 1004A

COMPUTATIONAL GRACEFUL DEGRADATION FOR VIDEO DECODING

Marco Mattavelli, Sylvain Brunetton
Integrated Systems Center - C3i,
Swiss Federal Institute of Technology,
Lausanne CH-1015, SWITZERLAND
Tel: +41 21 6936979; fax: +41 21 6934663
e-mail: Marco.Mattavelli@epfl.ch

ABSTRACT

The implementation of software based video decoders is an advantageous solution over traditional dedicated hardware real-time systems. However, guaranteeing real-time performance, needed when processing video/audio bit-streams, by scheduling processing tasks of variable load and that might exceed the available resources, is an extremely difficult and in many cases an impossible challenge. In this paper we introduce the motivations and ideas of the Computational Graceful Degradation (CGD) approach for the implementation of video decoding. The goal is to provide results of possibly lower visual quality, but respecting the real-time constraints, when the required processing needs exceed the available processing resources. This approach is very attractive since it enables a more efficient usage of the hardware processing power without needing to implement resources for worst case complexity decoding. We show how such technique can be applied to the decoding of compressed video sequences. Simulation results for MPEG-4 and H.263 video compression standards showing some of the interesting achievements and potentialities of CGD approach are also reported. The presented approach can be virtually applied to any video compression standard and processor based platform, although the best performances are achieved when complexity prediction information provided as side information by the video standard is available. This option has been included in the new MPEG-4 standard.

1 INTRODUCTION

The complexity of video decoding is a fluctuating function of time [1]. By complexity we mean the processing time necessary to decode a compressed video bitstream. Even considering a specific processor platform, a specific compression standard, with a relative level and profile, the complexity of video decoding is a variable quantity. Dedicated hardware platforms can guarantee worst case complexity decoding. In contrast, in the case of programmable platforms such a guarantee would imply the allocation of a processing power up to two orders of magnitude higher than the one actually needed for an average decoding [1]. Conversely, a theoretical average value is not useful for an efficient scheduling of a decoding task. The average value for each considered sequence is not stationary, and its variation around the theoretical average one are so unpredictable that no useful scheduling policy can be realized. The first straightforward option is a sustained over-sizing of the processing resources. The second is to accept interruptions in the audio/video output streams when the needed decoding resources exceed the ones available in the system. Obviously both options are not desirable being respectively not economical, or providing a quality of the services which may easily result perceptually unacceptable. The main rationale of the Computational Graceful Degradation (CGD) approach is to provide results of poorer quality when the results of the desirable quality cannot be obtained in real time. When the complexity of the task exceeds the available processing power, real-time services, possibly of degraded quality, are nevertheless guaranteed respecting all real-time constraints. This approach enables the correct scheduling of real-time tasks even when the computational system load fluctuates above the system capabilities.

The concept of computational graceful system degradation using imprecise results has previously been introduced in the field of real-time computing with applications to control system and automation. In these contexts imprecise results are defined as intermediate results with the property that the more recent and intermediate result is the closer one that approximates the final result (i.e. the idea only works in monotonically improving computations) [2, 3, 4]. In these cases the problems to be solved are more concerned with fault tolerance, I/O overloads, stability and reactivity to unexpected inputs, and the corresponding techniques are not suitable for the purposes of video decoding. In our case, the goal is to guarantee real-time video decoding with possible degradations of the image quality, i.e reductions of the peak signal to noise ratio (PSNR), without loosing parts of the image content. Such al-

Google Exhibit No. 1063
Page 0244

ternative is certainly much more desirable than other straightforward options such as cutting image portions or simply skipping frames. These latter options, as can be easily understood, can have catastrophic consequences on prediction based coding systems. In H.263 decoders, for instance, skipping the decoding of a single frame or a portion of it, implies the skipping of all future incoming frames (or portion of them) before the next incoming Intra frame. The time interval of such skipping operation might cover many seconds and the information content loss might be very relevant. Such abrupt interruptions, which are perceptually very annoying, might be caused not only by the simple fluctuation of the decoding complexity load, but also by the fluctuations of the load of other non-video processing tasks which reduce the available processing resources. Moreover, such events in general are out of the control of the video decoder system and cannot be easily forecasted. Considering these facts, it is evident the importance of techniques able to filter out complexity decoding peak loads or able to reduce the average decoding computation load without causing perceptually annoying consequences. Skipping B-VOP could be, in some cases, a viable option accepting the introduction of motion judder in the sequence reproduction. Unfortunately the presence of B-frames or B-VOP cannot be guaranteed in any bit-stream and just at the right time for which a reduction of the complexity is needed to respect the real time constraints.

In contrast with what could be commonly thought, also the classical multi-resolution coding features such as spatial or temporal scalabilities are not really useful for our goals. In general, they are used to optimize the transmission bandwidth usage enabling the compatibility of services at different titrates and spatial resolutions, but they must be embedded in the original bit-stream at the encoder side. If they are not embedded, the changes of resolution at the decoder size can results more computationally expansive than the simple decoding at the given resolution.

From these arguments the usefulness of CGD and its important features and applications are clear. The last key point that has left CGD as a good idea, but of difficult introduction in video coding, was that it was not clear how to implement it efficiently. Now that a technique able to predict with a good degree of approximation the required decoding time of each decoding process is demonstrated and available for MPEG-4 systems [5], we have the key for the useful and efficient implementation of CGD techniques.

2 PRINCIPLES OF DECODING USING THE CGD APPROACH

The decoding operations of a video compression algorithm can be divided into two main categories: 1) operations that cannot be simplified since they may prevent

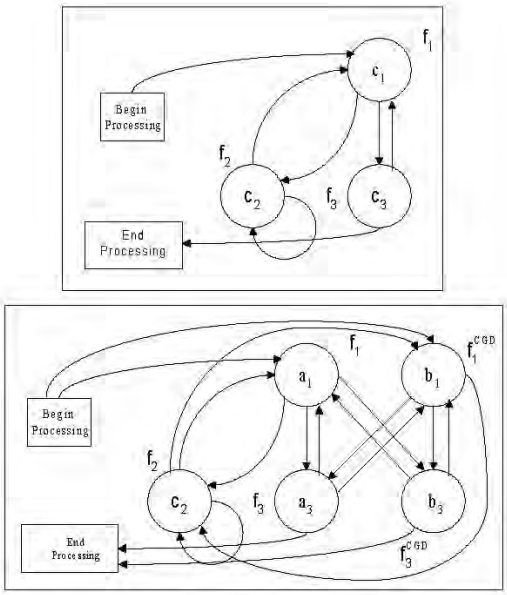


Figure 1: Upper diagram, example of a complexity model representing the data dependent execution of three algorithms. Lower diagram, example of the same complexity model of the upper diagram for which two algorithms (f_1 and f_3 are associated to new CGD algorithms (f_1^{CGD} and f_1^{CGD}).

the extraction and usage of relevant coding information, 2) operations that can be simplified, possibly leading to imprecise or degraded video results. To the first category typically belong all bit-stream parsing operations, entropy decoding (i.e. VLD and arithmetic coding), and all operations related to synchronization and to the syntax of the decoding process. For these operations no degradation mode definition is possible, otherwise it would imply skipping image portions or entire frames since either the data or the coding modes are lost. To the second category belong operations such as the IDCT, the prediction with and without motion compensation in its various forms (PB/B-frame prediction, overlapped block motion compensation (OBMC), half pixel interpolation, macroblock padding and so on). Simplifications of such operations applied to pixels or transform coefficients are always possible but at the expenses of imprecise results.

Two are the possible strategies for the implementation of CGD features. The first is to monitor the execution of the decoding process, to assume that the decoding time of each portion of the frame or VOP is proportional to the image area, and to execute CGD algorithms whenever the current monitored decoding time exceeds the current target decoding time suitable to respect the specified real-time constraints. Such simple strategy has several drawbacks. One is that there is no guaran-

Google Exhibit No. 1063
Page 0245

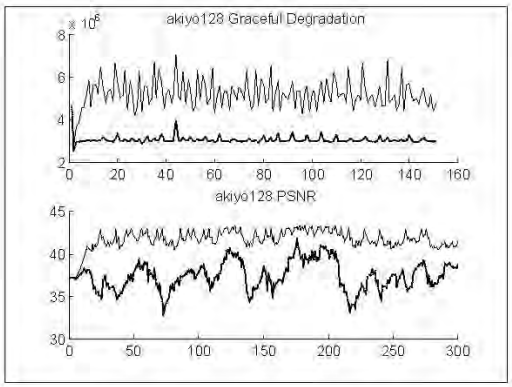


Figure 2: Example of CGD decoding of 300 frames of the sequence Akiyo at compressed at 128 kb/sec with H.263. The upper diagram reports the number of processor cycles for the original decoding (upper curve) and for the CGD decoding lower curve, versus the frame number. The lower diagram reports the two PSNR.

tee that the target decoding time is achieved. Another problem is due to execution of CGD algorithms that *a posteriori* could have not been necessary, thus introducing unnecessary image degradations. Moreover, such degradations are concentrated on critical image areas, and due to the drift given by the degraded reference for the prediction based algorithms, such degradations becomes visible and might seriously affect the resulting sequence PSNR. An example of such approach is reported in Fig. 2. The H.263 decoder has been modified by implementing computationally degraded IDCT and motion compensation algorithms. The interesting result achieved is that the decoding is obtained at a constant processing time for each frame. Unfortunately the PSNR degradations although limited to some dBs are nevertheless relevant from the perceptual point of view.

The second strategy is to exploit the complexity information carried by the frames or VOP headers, like it can be done for MPEG-4 systems [6], and use it in two stages. The first stage predicts the decoding time needed for the current frame or VOP, as described in [5]. The second, on the basis of the predicted decoding time and on the complexity information (i.e. statistics of the algorithms necessary to correctly decode the frame), executes only the CGD algorithms that are just necessary to not exceed the target decoding time.

More formally, let us consider a complexity model similar to the one illustrated in Fig. 1 composed by a set of I algorithms f_i, with $i = 1 \ldots I$, and suppose to be able to design for all, or only for a subset S^{CGD} of the functions f_i composing the model, new functions f_i^{CGD} which have the following three properties: 1. the execution of a function f_i^{CGD} yields results that are an approximation of the ones produced by the original function f_i, 2. the execution of f_i^{CGD} does not prevent the execution of any subsequent decoding functions, 3. the weight w_i^{CGD} associated to f_i^{CGD} that represent the execution time of the f_i^{CGD} algorithm for the considered decoder implementation is lower than the corresponding w_i associated to f_i (i.e. $w_i^{CGD} < w_i$). A schematic representation of a complexity model which includes new degraded functions is reported in Fig. 1. The picture shows that new degraded algorithms can be executed at the place of the original algorithms. Obviously, the execution of any f_i^{CGD} may produce PSNR degradations in the reconstruction of some image parts, it does not prevent from correctly decoding the rest of the frame or VOP, but it reduces the necessary decoding time. Let us partition the functions constituting the complexity model into two disjoint sets: S^{Orig} and S^{CGD} respectively for the ones that do not admit a CGD version (f_2 for instance in Fig. 1) and for the ones for which it is possible to design a CGD version (f_1 and f_3). Let us consider for simplicity an image sequence composed by a sequence of single frames or VOPs and suppose that the execution time for the decoding of the frame or VOP is larger than the allocated decoding time:

$$T(t) > T_{Alloc}, \tag{1}$$

where T(t) is the decoding time of the VOP at hand at the sampling time t. Since the decoding time can be approximated by its prediction [5] it can be expressed by:

$$T(t) \simeq T_{Pred}(t) = \sum_{i=1}^{I} c_i w_i, \tag{2}$$

where the c_i are the actual occurrences of algorithm f_i in the current frame or VOP [5], and the w_i are the corresponding complexity weights. The lower limit for the decoding time executing all CGD algorithms f_i^{CGD} at the place of f_i for all $i \in S^{CGD}$, is given by:

$$T_{CGD} = \sum_{i \in S_{CGD}} c_i w_i^{CGD} + \sum_{i \in S_{Orig}} c_i w_i. \tag{3}$$

If $T_{CGD} < T_{Alloc} < T_{Pred}$, is possible to guarantee the real-time constraints by executing the f_i^{CGD} algorithms and accepting approximate decoding results. It is also possible to find for each c_i with $i \in S^{CGD}$, a couple of coefficients a_i, b_i, for which $a_i + b_i = c_i$, and denoting as $T_{Process}$ any decoding time interval which respects the real time constraints, we can express the desired process decoding time as any decoding time value that satisfy the expression:

$$T_{process} = \sum_{i \in S_{CGD}} (a_i w_i^{CGD} + b_i w_i) + \sum_{i \in S_{Orig}} c_i w_i \leq T_{Alloc}. \tag{4}$$

Obviously for any Allocated Decoding Time the goal is to respect the real-time constraints with the lowest

Google Exhibit No. 1063
Page 0246

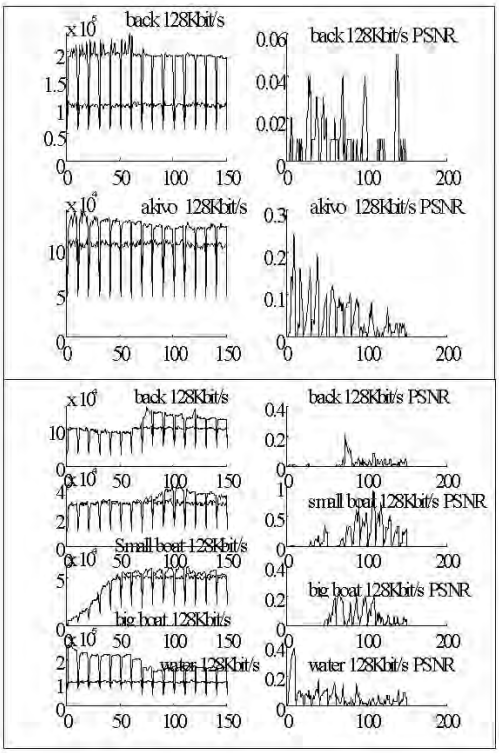


Figure 3: Left, decoding time results for the various VOP of sequences Akiyo and Coastguard, with and without imposing a bound. Right, the PSNR degradations versus the sequence VOP number.

degradation in the video/audio content. Associating a degradation measure to each w_i^{CGD} it is possible to minimize the degradation for any given T_{Alloc}. The challenge is to find the set $S_{a,b}$ of couple of coefficients a_i, b_i, with $i \in S_{CGD}$, that satisfy minimize the video degradations expressed as difference with the non degraded output:

$$S_{a,b} = \text{argmin(a, b)}_{\text{i}\in S_{CGD}} : (\text{PSNR} - \text{PSNR(a, b)}). \quad (5)$$

The determination of the set $S_{a,b}$ can be done by means of many suitable optimization method or heuristics. The described techniques can be easily extended to the case of sequences containing multiple VOP at the same sampling time t.

3 SIMULATION RESULTS

Simulation results demonstrate the possibility of successful implementations of CGD for MPEG-4 video decoding. A bound on the decoding complexity of each VOP is defined setting a maximum constant T_{Alloc}. Figure 3 report the results of MPEG-4 video decoding imposing such bounds on the decoding time of each video object plane (VOP) for the test images Coastguard and Akiyo coded at 128 kb/s. On the left, the graphs report the decoding time in msec for the unbounded and bounded decoding (lower curve) versus the image sequence number. On the right the graphs reports the resulting PSNR degradation. The decoder is the Momusys CD on a Sun Ultrasparc station modified implementing degraded motion compensation algorithms. Obviously the actual results depend on the decoder, on the platform on the CGD algorithms implemented, but the good results demonstrate that bounded complexity decodings are possible with minimal image degradations much lower than the one obtained without complexity information and decoding time prediction.

4 CONCLUSION

We have introduced the motivations and concepts of the CGD approach for the implementation of video decoders on processor based platforms. Simulation results for MPEG-4 video decoding demonstrate the successful implementation of the CGD approach. Obviously, the implementation of CGD features are not proposed to be used at their maximum degradation rate but as tool able to guarantee real-time performance in presence of peaks of decoding complexity or as a way to ease the scheduling problems given by temporary fluctuations of the available processing power. Moreover, the presented approach may enable the successful implementation of video decoding for levels and profiles that would be forbidden if the complexity of *worst case decoding* has to be guaranteed.

References

[1] M. Mattavelli and S. Brunetton. Scalability of the mpeg-4 vm based on the computational power of texture decoding. Technical Report Document M925, ISO/IEC JTC1/SC29/WG11, Tampere, Finland, July 1996.

[2] A.D. Stoyenko W.A. Halang. *Constructing predictable real time systems.* Kluwer Academic Publisher, 1991.

[3] J. Liu K. Lin, S. Natarjan. Imprecise results: Utilizing partial computations in real-time systems. In *Proc. of IEEE Real-Time System Symp.*, Dec. 1987.

[4] S. Natarjan J. Liu, K. Lin. Scheduling real time periodic jobs using imprecise results. In *Proc. of IEEE Real-Time System Symp.*, Dec. 1987.

[5] M. Mattavelli and S. Brunetton. Prediction of decoding time for mpeg-4 video. In *Proc. of Eusipco 98, IX European Signal Processing Conference*, Island of Rhodes, Greece, September 8-11 1998.

[6] ISO-IEC/JTC1. Doc for iso/iec cd 14496-2 visual. Technical Report Doc. W2119, Tokyo, March 1998.

Google Exhibit No. 1063
Page 0247

Ex. 1004B

LIBRARY | CATALOG

1 of 1

BOOK

Signal processing IX : theories and applications : proceedings of EUSIPCO-98, ...

Full Record MARC Tags

Meeting name

EUSIPCO-98 (1998 : Rhodes, Greece)

Main title

Signal processing IX : theories and applications : proceedings of EUSIPCO-98, Ninth European Signal Processing Conference, Rhodes, Greece, 8-11 September 1998 / edited by S. Theodoridis ... [et al.].

Published/Created

Patras, Greece : Typorama Editions, 1998.

Request this Item | LC Find It

More Information

LCCN Permalink	https://lccn.loc.gov/98232284
Description	4 v. (lxii, 2568 p.) : ill. ; 28 cm. + 1 computer laser optical disc (4 3/4 in.)
ISBN	9607620054 (set)
LC classification	TK5102.9 .E97 1998

Google Exhibit No. 1063
Page 0249

Variant title	Signal processing 9 Signal processing nine
Related names	Theodoridis, Sergios, 1951-
LC Subjects	Signal processing--Congresses.
Browse by shelf order	TK5102.9
Notes	Includes bibliographical references and indexes.
LCCN	98232284
Dewey class no.	621.382/2
Type of material	Book

Item Availability

CALL NUMBER	TK5102.9 .E97 1998 Copy 1
Request in	Jefferson or Adams Building Reading Rooms
Status	Not Charged

Google Exhibit No. 1063
Page 0250

LIBRARY | CATALOG

1 of 1

BOOK

Signal processing IX : theories and applications : proceedings of EUSIPCO-98, ...

Full Record **MARC Tags**

000		01234nam a2200289 a 4500
001		1682236
005		19990122143936.3
008		981216s1998 gr a b 101 0 eng
035	__	**|9** (DLC) 98232284
906	__	**|a** 7 **|b** cbc **|c** orignew **|d** 2 **|e** ncip **|f** 19 **|g** y-gencatlg
955	__	**|a** OK03 12-16-98 to Cat.; jf00 12-23-98; jf06 to subj 01-07-99; jf08 01-07-99 to SL; jf25 01-08-99 to ddc
010	__	**|a** 98232284
020	__	**|a** 9607620054 (set)
040	__	**|a** DLC **|c** DLC
050	00	**|a** TK5102.9 **|b** .E97 1998
082	00	**|a** 621.382/2 **|2** 21
111	2_	**|a** EUSIPCO-98 **|d** (1998 : **|c** Rhodes, Greece)
245	10	**|a** Signal processing IX : **|b** theories and applications : proceedings of EUSIPCO-98, Ninth European Signal Processing Conference, Rhodes, Greece, 8-11 September 1998 / **|c** edited by S. Theodoridis ... [et al.].
246	3_	**|a** Signal processing 9
246	3_	**|a** Signal processing nine
260	__	**|a** Patras, Greece : **|b** Typorama Editions, **|c** 1998.
300	__	**|a** 4 v. (lxii, 2568 p.) : **|b** ill. ; **|c** 28 cm. + **|e** 1 computer laser optical disc (4 3/4 in.)
504	__	**|a** Includes bibliographical references and indexes.

Google Exhibit No. 1063
Page 0251

650	_0	|a Signal processing |x Congresses.
700	1_	|a Theodoridis, Sergios, |d 1951-
922	__	|a ag
991	__	|b c-GenColl |h TK5102.9 |i .E97 1998 |t Copy 1 |w BOOKS

Request this Item | LC Find It

Item Availability

CALL NUMBER	TK5102.9 .E97 1998 Copy 1
Request in	Jefferson or Adams Building Reading Rooms
Status	Not Charged

Google Exhibit No. 1063
Page 0252

Ex. 1004C

libraries.org

A directory of libraries throughout the world

found **3** items where the item contains **Library of Congress**. Showing item **2** of **3**.

Library of Congress

United States -- Library of Congress

Address: 101 Independence Avenue SE
Washington, District of Columbia
20540
United States

Phone: 202-707-5000

Connect to: **Library Web Site** ✔ **Online Catalog**

Library details: **Library of Congress** is a **National** library.
This library is affiliated with **Library of Congress (view map)** . The collection of the library contains **175,000,000** volumes. The library circulates **250,000** items per year.

Library Hours		
Day	Open	Close
Sunday	Closed	
Monday	8:30 am	8:30 pm
Tuesday	8:30 am	8:30 pm
Wednesday	8:30 am	8:30 pm
Thursday	8:30 am	8:30 pm
Friday	8:30 am	5:00 pm
Saturday	8:30 am	5:00 pm



Library of Congress



map location ✔

Other Info: Went live with Voyager on August 15, 1999. Currently planning to implement the Library Collections Access Platform (LCAP) on FOLIO and EBSCO discovery services in FY25.

Permalink: **https://librarytechnology.org/library/2263**
(Use this link to refer back to this listing.)

Administration: The director of the library is **Dr. Carla Hayden** .

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

ARL

Affiliations: The library is a member of the **Association of Research Libraries**

Collection size	175,000,000	**volumes**
Annual Circulation	250,000	**transactions**

Wireless: The library offers **wireless** access to the Internet. http://www.loc.gov/rr/wirelessaccess.html

Related Libraries

- All members of the **Association of Research Libraries** members
- **Directory of ARL Member Libraries**
- Libraries located in **Washington** (District of Columbia)
- **all National libs in District of Columbia**

About libraries.org

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

This directory is a component of **Library Technology Guides**.

Google Exhibit No. 1063
Page 0254

Technology Profile		
	Product Name	Year Contracted
Current Automation System	FOLIO -- EBSCO Information Services	2023
Previous Automation System	Voyager	1998
Discovery Service (w/index)	Ex Libris CDI	
Electronic Resource Management	Electronic Resource Management	
Item ID Type	Barcode	
This library is responsible for the procurement of the library automation system.		

View Privacy and Security Report

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 **FOLIO -- EBSCO Information Services**.

Respond to the 2024 Library Automation Perceptions Survey

Identifiers	
libraries.org ID	2263
OCLC Symbol	DLC
WorldCat Registry ID	45266

Record History: This listing was created on Aug 9, 1998 and was last modified on Feb 13, 2025.

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**.

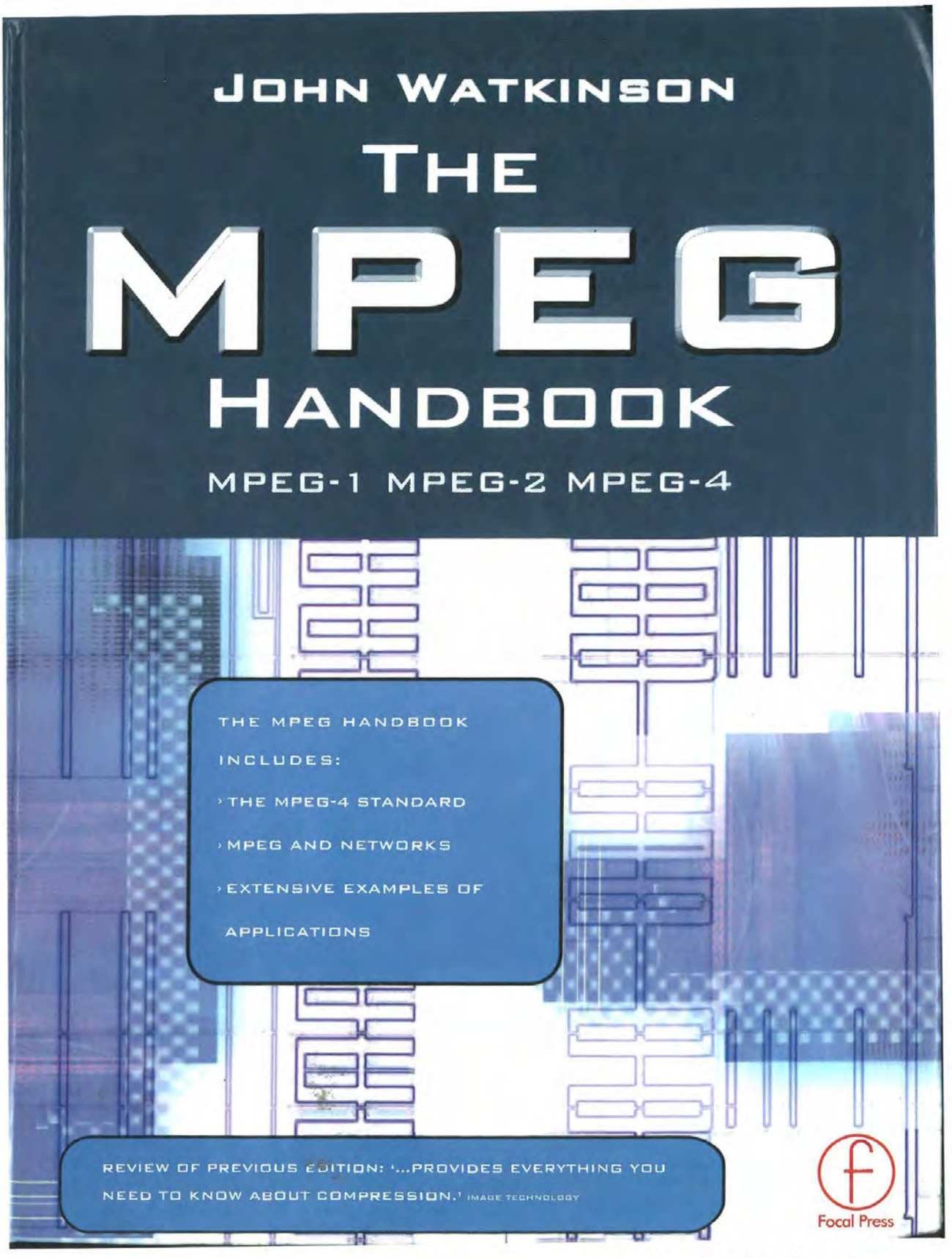
Please **contact Marshall Breeding** to report corrections about information listed for this library.

libraries.org | Update this Entry | Library Technology Guides

Maintained by **Marshall Breeding**

terms of service
Copyright 1994-2024

Google Exhibit No. 1063
Page 0255



Google Exhibit No. 1005
Page 001

Google Exhibit No. 1063
Page 0256

The MPEG Handbook

MPEG-1, MPEG-2, MPEG-4

John Watkinson



Focal Press
OXFORD AUCKLAND BOSTON JOHANNESBURG MELBOURNE NEW DELHI

Google Exhibit No. 1005
Page 002

Google Exhibit No. 1063
Page 0257

Focal Press
An imprint of Butterworth-Heinemann
Linacre House, Jordan Hill, Oxford OX2 8DP
225 Wildwood Avenue, Woburn, MA 01801-2041
A division of Reed Educational and Professional Publishing Ltd

A member of the Reed Elsevier plc group

First published 2001

© John Watkinson 2001

All rights reserved. No part of this publication may be reproduced in any material form (including photocopying or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication) without the written permission of the copyright holder except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 0LP. Applications for the copyright holder's written permission to reproduce any part of this publication should be addressed to the publishers

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication Data
A catalogue record for this book is available from the Library of Congress

For information on all Focal Press publications visit our website at www.focalpress.com

ISBN 0 240 51656 7

Composition by Genesis Typesetting, Rochester, Kent
Printed and bound in Great Britain

PLANT A TREE
BTCV British Trust for Conservation Volunteers
FOR EVERY TITLE THAT WE PUBLISH, BUTTERWORTH-HEINEMANN WILL PAY FOR BTCV TO PLANT AND CARE FOR A TREE.

Contents

Preface		xi
Acknowledgements		xiii
Chapter 1	**Introduction to compression**	**1**
1.1	What is MPEG?	1
1.2	Why compression is necessary	3
1.3	MPEG-1, 2 and 4 contrasted	4
1.4	Some applications of compression	5
1.5	Lossless and perceptive coding	6
1.6	Compression principles	8
1.7	Video compression	13
	1.7.1 Intra-coded compression	14
	1.7.2 Inter-coded compression	15
	1.7.3 Introduction to motion compensation	16
	1.7.4 Film-originated video compression	18
1.8	Introduction to MPEG-1	19
1.9	MPEG-2: Profiles and Levels	20
1.10	Introduction to MPEG-4	22
1.11	Audio compression	26
	1.11.1 Sub-band coding	26
	1.11.2 Transform coding	26
	1.11.3 Predictive coding	26
1.12	MPEG bitstreams	27
1.13	Drawbacks of compression	28

Google Exhibit No. 1005
Page 004

Google Exhibit No. 1063
Page 0259

1.14 Compression pre-processing 29
1.15 Some guidelines 30
References 31

Chapter 2 Fundamentals 32

2.1 What is an audio signal? 32
2.2 What is a video signal? 32
2.3 Types of video 33
2.4 What is a digital signal? 35
2.5 Sampling 38
2.6 Reconstruction 42
2.7 Aperture effect 46
2.8 Choice of audio sampling rate 49
2.9 Video sampling structures 51
2.10 The phase-locked loop 54
2.11 Quantizing 56
2.12 Quantizing error 58
2.13 Dither 61
2.14 Introduction to digital processing 63
2.15 Logic elements 65
2.16 Storage elements 67
2.17 Binary coding 69
2.18 Gain control 78
2.19 Floating-point coding 81
2.20 Multiplexing principles 82
2.21 Packets 83
2.22 Statistical multiplexing 84
2.23 Timebase correction 84
References 86

Chapter 3 Processing for compression 87

3.1 Introduction 87
3.2 Transforms 90
3.3 Convolution 92
3.4 FIR and IIR filters 95
3.5 FIR filters 95
3.6 Interpolation 102
3.7 Downsampling filters 113
3.8 The quadrature mirror filter 114
3.9 Filtering for video noise reduction 118
3.10 Warping 119

Google Exhibit No. 1005
Page 005

Google Exhibit No. 1063
Page 0260

3.11	Transforms and duality	126
3.12	The Fourier transform	128
3.13	The discrete cosine transform (DCT)	137
3.14	The wavelet transform	138
3.15	The importance of motion compensation	144
3.16	Motion-estimation techniques	146
3.16.1	Block matching	146
3.16.2	Gradient matching	148
3.16.3	Phase correlation	149
3.17	Motion-compensated displays	153
3.18	Camera-shake compensation	155
3.19	Motion-compensated de-interlacing	157
3.20	Compression and requantizing	158
References		163

Chapter 4 Audio compression **165**

4.1	Introduction	165
4.2	The deciBel	166
4.3	Audio level metering	171
4.4	The ear	174
4.5	The cochlea	176
4.6	Level and loudness	178
4.7	Frequency discrimination	179
4.8	Critical bands	180
4.9	Beats	182
4.10	Codec level calibration	184
4.11	Quality measurement	185
4.12	The limits	187
4.13	Compression applications	187
4.14	Audio compression tools	188
4.15	Sub-band coding	193
4.16	Audio compression formats	194
4.17	MPEG audio compression	194
4.18	MPEG Layer I audio coding	197
4.19	MPEG Layer II audio coding	202
4.20	MPEG Layer III audio coding	204
4.21	MPEG-2 AAC – advanced audio coding	207
4.22	Dolby AC-3	213
4.23	MPEG-4 Audio	215
4.24	MPEG-4 AAC	215
4.25	Compression in stereo and surround sound	216
References		222

Google Exhibit No. 1005
Page 006

Google Exhibit No. 1063
Page 0261

Chapter 5 MPEG video compression 224

5.1 The eye 224
5.2 Dynamic resolution 228
5.3 Contrast 232
5.4 Colour vision 233
5.5 Colour difference signals 235
5.6 Progressive or interlaced scan? 238
5.7 Spatial and temporal redundancy in MPEG 243
5.8 *I* and *P* coding 248
5.9 Bidirectional coding 249
5.10 Coding applications 252
5.11 Intra-coding 253
5.12 Intra-coding in MPEG-1 and MPEG-2 257
5.13 A bidirectional coder 261
5.14 Slices 265
5.15 Handling interlaced pictures 266
5.16 MPEG-1 and MPEG-2 coders 271
5.17 The elementary stream 272
5.18 An MPEG-2 decoder 273
5.19 MPEG-4 276
5.20 Video objects 278
5.21 Texture coding 281
5.22 Shape coding 287
5.23 Padding 289
5.24 Video object coding 290
5.25 Two-dimensional mesh coding 293
5.26 Sprites 298
5.27 Wavelet-based compression 300
5.28 Three-dimensional mesh coding 304
5.29 Animation 313
5.30 Scaleability 315
5.31 Coding artifacts 318
5.32 MPEG and concatenation 322
References 328

Chapter 6 Program and transport streams 329

6.1 Introduction 329
6.2 Packets and time stamps 329
6.3 Transport streams 332
6.4 Clock references 333
6.5 Program Specific Information (PSI) 335

Google Exhibit No. 1005
Page 007

Google Exhibit No. 1063
Page 0262

6.6 Multiplexing 336
6.7 Remultiplexing 338
Reference 339

Chapter 7 MPEG applications 340

7.1 Introduction 340
7.2 Video phones 342
7.3 Digital television broadcasting 342
7.4 The DVB receiver 355
7.5 CD-Video and DVD 356
7.6 Personal video recorders 361
7.7 Networks 363
7.8 FireWire 371
7.9 Broadband networks and ATM 373
7.10 ATM AALs 377
References 380

Index 381

Google Exhibit No. 1005
Page 008

Google Exhibit No. 1063
Page 0263

1

Introduction to compression

1.1 What is MPEG?

MPEG is actually an acronym for the Moving Pictures Experts Group which was formed by the ISO (International Standards Organization) to set standards for audio and video compression and transmission.

Compression is summarized in Figure 1.1. It will be seen in (a) that the data rate is reduced at source by the *compressor*. The compressed data are then passed through a communication channel and returned to the original rate by the *expander*. The ratio between the source data rate and the channel data rate is called the *compression factor*. The term *coding gain* is also used. Sometimes a compressor and expander in series are referred to as a *compander*. The compressor may equally well be referred to as a *coder* and the expander a *decoder* in which case the tandem pair may be called a *codec*.

Where the encoder is more complex than the decoder, the system is said to be asymmetrical. Figure 1.1(b) shows that MPEG works in this way. The encoder needs to be algorithmic or adaptive whereas the decoder is 'dumb' and carries out fixed actions. This is advantageous in applications such as broadcasting where the number of expensive complex encoders is small but the number of simple inexpensive decoders is large. In point-to-point applications the advantage of asymmetrical coding is not so great.

The approach of the ISO to standardization in MPEG is novel because it is not the encoder which is standardized. Figure 1.2(a) shows that instead the way in which a decoder shall interpret the bitstream is defined. A decoder which can successfully interpret the bitstream is said to be *compliant*. Figure 1.2(b) shows that the advantage of standardizing the decoder is that over time encoding algorithms can improve yet compliant decoders will continue to function with them.

Google Exhibit No. 1005
Page 009

Google Exhibit No. 1063
Page 0264

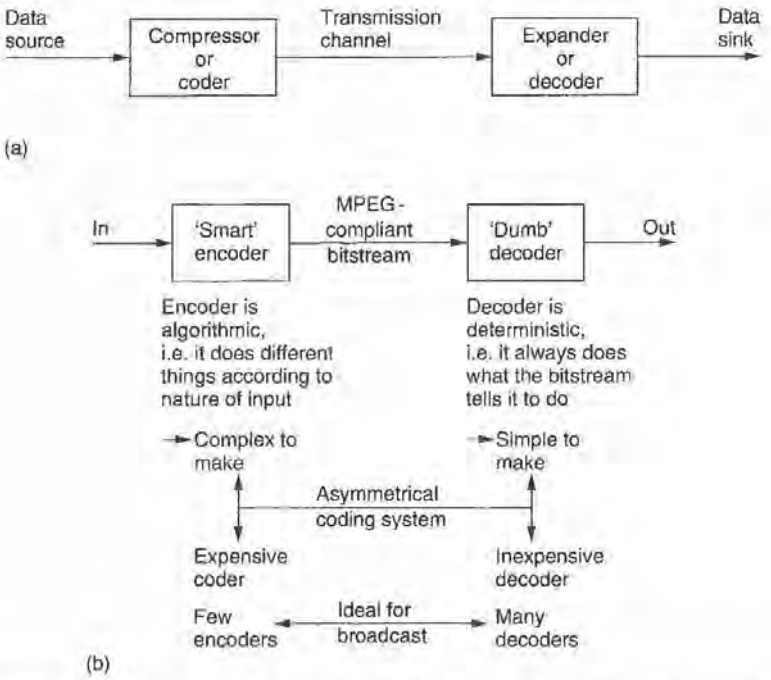


Figure 1.1 In (a) a compression system consists of compressor or coder, a transmission channel and a matching expander or decoder. The combination of coder and decoder is known as a codec. (b) MPEG is asymmetrical since the encoder is much more complex than the decoder.

It should be noted that a compliant decoder must correctly be able to interpret every allowable bitstream, whereas an encoder which produces a restricted subset of the possible codes can still be compliant.

The MPEG standards give very little information regarding the structure and operation of the encoder. Provided the bitstream is compliant, any coder construction will meet the standard, although some designs will give better picture quality than others. Encoder construction is not revealed in the bitstream and manufacturers can supply encoders using algorithms which are proprietary and their details do not need to be published. A useful result is that there can be competition between different encoder designs which means that better designs can evolve. The user will have greater choice because different levels of cost and complexity can exist in a range of coders yet a compliant decoder will operate with them all.

MPEG is, however, much more than a compression scheme as it also standardizes the protocol and syntax under which it is possible to combine or multiplex audio data with video data to produce a digital equivalent of a television program. Many such programs can be

Google Exhibit No. 1005
Page 010

Google Exhibit No. 1063
Page 0265

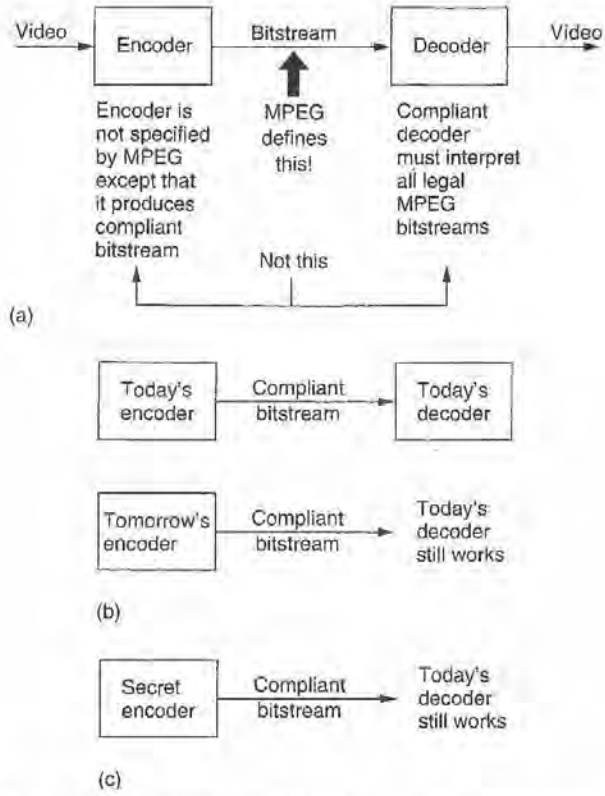


Figure 1.2 (a) MPEG defines the protocol of the bitstream between encoder and decoder. The decoder is defined by implication, the encoder is left very much to the designer. (b) This approach allows future encoders of better performance to remain compatible with existing decoders. (c) This approach also allows an encoder to produce a standard bitstream while its technical operation remains a commercial secret.

combined in a single multiplex and MPEG defines the way in which such multiplexes can be created and transported. The definitions include the metadata which decoders require to demultiplex correctly and which users will need to locate programs of interest.

As with all video systems there is a requirement for synchronizing or genlocking and this is particularly complex when a multiplex is assembled from many signals which are not necessarily synchronized to one another.

1.2 Why compression is necessary

Compression, bit rate reduction, data reduction and source coding are all terms which mean basically the same thing in this context. In essence the

Google Exhibit No. 1005
Page 011

Google Exhibit No. 1063
Page 0266

same (or nearly the same) information is carried using a smaller quantity or rate of data. It should be pointed out that in audio *compression* traditionally means a process in which the dynamic range of the sound is reduced. In the context of MPEG the same word means that the bit rate is reduced, ideally leaving the dynamics of the signal unchanged. Provided the context is clear, the two meanings can co-exist without a great deal of confusion.

There are several reasons why compression techniques are popular:

(a) Compression extends the playing time of a given storage device.
(b) Compression allows miniaturization. With fewer data to store, the same playing time is obtained with smaller hardware. This is useful in ENG (electronic news gathering) and consumer devices.
(c) Tolerances can be relaxed. With fewer data to record, storage density can be reduced making equipment which is more resistant to adverse environments and which requires less maintenance.
(d) In transmission systems, compression allows a reduction in bandwidth which will generally result in a reduction in cost. This may make possible a service which would be impracticable without it.
(e) If a given bandwidth is available to an uncompressed signal, compression allows faster than real-time transmission in the same bandwidth.
(f) If a given bandwidth is available, compression allows a better-quality signal in the same bandwidth.

1.3 MPEG-1, 2 and 4 contrasted

The first compression standard for audio and video was MPEG-1.[1,2] Although many applications have been found, MPEG-1 was basically designed to allow moving pictures and sound to be encoded into the bit rate of an audio Compact Disc. The resultant Video-CD was quite successful but has now been superseded by DVD. In order to meet the low bit requirement, MPEG-1 downsampled the images heavily as well as using picture rates of only 24–30 Hz and the resulting quality was moderate.

The subsequent MPEG-2 standard was considerably broader in scope and of wider appeal.[3] For example, MPEG-2 supports interlace and HD whereas MPEG-1 did not. MPEG-2 has become very important because it has been chosen as the compression scheme for both DVB (digital video broadcasting) and DVD (digital video disk). Developments in standardizing scaleable and multi-resolution compression which would have become MPEG-3 were ready by the time MPEG-2 was ready to be

Google Exhibit No. 1005
Page 012

Google Exhibit No. 1063
Page 0267

standardized and so this work was incorporated into MPEG-2, and as a result there is no MPEG-3 standard.

MPEG-4[4] uses further coding tools with additional complexity to achieve higher compression factors than MPEG-2. In addition to more efficient coding of video, MPEG-4 moves closer to computer graphics applications. In the more complex Profiles, the MPEG-4 decoder effectively becomes a rendering processor and the compressed bitstream describes three-dimensional shapes and surface texture. It is to be expected that MPEG-4 will become as important to Internet and wireless delivery as MPEG-2 has become in DVD and DVB.

1.4 Some applications of compression

The applications of audio and video compression are limitless and the ISO has done well to provide standards which are appropriate to the wide range of possible compression products.

MPEG coding embraces video pictures from the tiny screen of a videophone to the high-definition images needed for electronic cinema. Audio coding stretches from speech-grade mono to multichannel surround sound.

Figure 1.3 shows the use of a codec with a recorder. The playing time of the medium is extended in proportion to the compression factor. In the case of tapes, the access time is improved because the length of tape needed for a given recording is reduced and so it can be rewound more quickly. In the case of DVD (digital video disk aka digital versatile disk) the challenge was to store an entire movie on one 12 cm disk. The storage density available with today's optical disk technology is such that consumer recording of conventional uncompressed video would be out of the question.

In communications, the cost of data links is often roughly proportional to the data rate and so there is simple economic pressure to use a high compression factor. However, it should be borne in mind that implement-

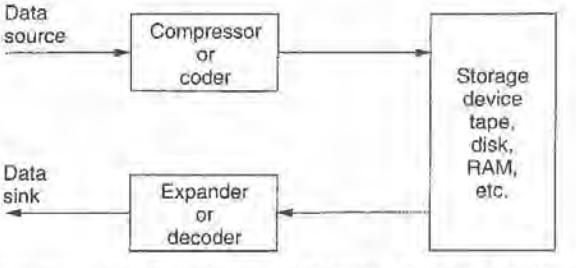


Figure 1.3 Compression can be used around a recording medium. The storage capacity may be increased or the access time reduced according to the application.

Google Exhibit No. 1005
Page 013

Google Exhibit No. 1063
Page 0268

ing the codec also has a cost which rises with compression factor and so a degree of compromise will be inevitable.

In the case of video-on-demand, technology exists to convey full bandwidth video to the home, but to do so for a single individual at the moment would be prohibitively expensive. Without compression, HDTV (high-definition television) requires too much bandwidth. With compression, HDTV can be transmitted to the home in a similar bandwidth to an existing analog SDTV channel. Compression does not make video-on-demand or HDTV possible; it makes them economically viable.

In workstations designed for the editing of audio and/or video, the source material is stored on hard disks for rapid access. Whilst top-grade systems may function without compression, many systems use compression to offset the high cost of disk storage. In some systems a compressed version of the top-grade material may also be stored for browsing purposes.

When a workstation is used for *off-line* editing, a high compression factor can be used and artifacts will be visible in the picture. This is of no consequence as the picture is only seen by the editor who uses it to make an EDL (edit decision list) which is no more than a list of actions and the timecodes at which they occur. The original uncompressed material is then *conformed* to the EDL to obtain a high-quality edited work. When *on-line* editing is being performed, the output of the workstation is the finished product and clearly a lower compression factor will have to be used. Perhaps it is in broadcasting where the use of compression will have its greatest impact. There is only one electromagnetic spectrum and pressure from other services such as cellular telephones makes efficient use of bandwidth mandatory. Analog television broadcasting is an old technology and makes very inefficient use of bandwidth. Its replacement by a compressed digital transmission is inevitable for the practical reason that the bandwidth is needed elsewhere.

Fortunately in broadcasting there is a mass market for decoders and these can be implemented as low-cost integrated circuits. Fewer encoders are needed and so it is less important if these are expensive. Whilst the cost of digital storage goes down year on year, the cost of the electromagnetic spectrum goes up. Consequently in the future the pressure to use compression in recording will ease whereas the pressure to use it in radio communications will increase.

1.5 Lossless and perceptive coding

Although there are many different coding techniques, all of them fall into one or other of these categories. In *lossless* coding, the data from the expander are identical bit-for-bit with the original source data. The

Google Exhibit No. 1005
Page 014

Google Exhibit No. 1063
Page 0269

so-called 'stacker' programs which increase the apparent capacity of disk drives in personal computers use lossless codecs. Clearly with computer programs the corruption of a single bit can be catastrophic. Lossless coding is generally restricted to compression factors of around 2:1.

It is important to appreciate that a lossless coder cannot guarantee a particular compression factor and the communications link or recorder used with it must be able to function with the variable output data rate. Source data which result in poor compression factors on a given codec are described as *difficult*. It should be pointed out that the difficulty is often a function of the codec. In other words data which one codec finds difficult may not be found difficult by another. Lossless codecs can be included in bit-error-rate testing schemes. It is also possible to cascade or *concatenate* lossless codecs without any special precautions.

Higher compression factors are only possible with *lossy* coding in which data from the expander are not identical bit-for-bit with the source data and as a result comparing the input with the output is bound to reveal differences. Lossy codecs are not suitable for computer data, but are used in MPEG as they allow greater compression factors than lossless codecs. Successful lossy codecs are those in which the errors are arranged so that a human viewer or listener finds them subjectively difficult to detect. Thus lossy codecs must be based on an understanding of psycho-acoustic and psycho-visual perception and are often called *perceptive* codes.

In perceptive coding, the greater the compression factor required, the more accurately must the human senses be modelled. Perceptive coders can be forced to operate at a fixed compression factor. This is convenient for practical transmission applications where a fixed data rate is easier to handle than a variable rate. The result of a fixed compression factor is that the subjective quality can vary with the 'difficulty' of the input material. Perceptive codecs should not be concatenated indiscriminately especially if they use different algorithms. As the reconstructed signal from a perceptive codec is not bit-for-bit accurate, clearly such a codec cannot be included in any bit error rate testing system as the coding differences would be indistinguishable from real errors.

Although the adoption of digital techniques is recent, compression itself is as old as television. Figure 1.4 shows some of the compression techniques used in traditional television systems.

Most video signals employ a non-linear relationship between brightness and the signal voltage which is known as gamma. Gamma is a perceptive coding technique which depends on the human sensitivity to video noise being a function of the brightness. The use of gamma allows the same subjective noise level with an eight-bit system as would be achieved with a fourteen-bit linear system.

One of the oldest techniques is interlace, which has been used in analog television from the very beginning as a primitive way of reducing

Google Exhibit No. 1005
Page 015

Google Exhibit No. 1063
Page 0270

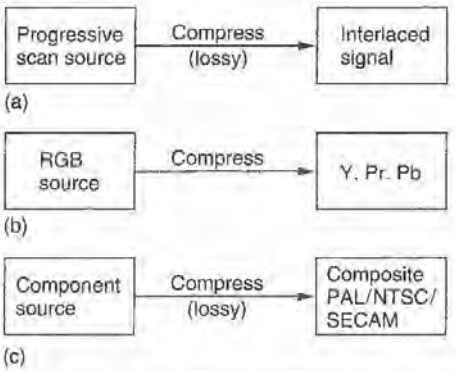


Figure 1.4 Compression is as old as television. (a) Interlace is a primitive way of halving the bandwidth. (b) Colour difference working invisibly reduces colour resolution. (c) Composite video transmits colour in the same bandwidth as monochrome.

bandwidth. As will be seen in Chapter 5, interlace is not without its problems, particularly in motion rendering. MPEG-2 supports interlace simply because legacy interlaced signals exist and there is a requirement to compress them. This should not be taken to mean that it is a good idea.

The generation of colour difference signals from *RGB* in video represents an application of perceptive coding. The human visual system (HVS) sees no change in quality although the bandwidth of the colour difference signals is reduced. This is because human perception of detail in colour changes is much less than in brightness changes. This approach is sensibly retained in MPEG.

Composite video systems such as PAL, NTSC and SECAM are all analog compression schemes which embed a subcarrier in the luminance signal so that colour pictures are available in the same bandwidth as monochrome. In comparison with a linear-light progressive scan *RGB* picture, gamma-coded interlaced composite video has a compression factor of about 10:1.

In a sense MPEG-2 can be considered to be a modern digital equivalent of analog composite video as it has most of the same attributes. For example, the eight-field sequence of the PAL subcarrier which makes editing difficult has its equivalent in the GOP (group of pictures) of MPEG.

1.6 Compression principles

In a PCM digital system the bit rate is the product of the sampling rate and the number of bits in each sample and this is generally constant.

Google Exhibit No. 1005
Page 016

Google Exhibit No. 1063
Page 0271

Nevertheless the *information* rate of a real signal varies. In all real signals, part of the signal is obvious from what has gone before or what may come later and a suitable receiver can predict that part so that only the true information actually has to be sent. If the characteristics of a predicting receiver are known, the transmitter can omit parts of the message in the knowledge that the receiver has the ability to re-create it. Thus all encoders must contain a model of the decoder.

One definition of information is that it is the unpredictable or surprising element of data. Newspapers are a good example of information because they only mention items which are surprising. Newspapers never carry items about individuals who have *not* been involved in an accident as this is the normal case. Consequently the phrase 'no news is good news' is remarkably true because if an information channel exists but nothing has been sent then it is most likely that nothing remarkable has happened.

The unpredictability of the punch line is a useful measure of how funny a joke is. Often the build-up paints a certain picture in the listener's imagination, which the punch line destroys utterly. One of the author's favourites is the one about the newly married couple who didn't know the difference between putty and petroleum jelly – their windows fell out.

The difference between the information rate and the overall bit rate is known as the redundancy. Compression systems are designed to eliminate as much of that redundancy as practicable or perhaps affordable. One way in which this can be done is to exploit statistical predictability in signals. The information content or *entropy* of a sample is a function of how different it is from the predicted value. Most signals have some degree of predictability. A sine wave is highly predictable, because all cycles look the same. According to Shannon's theory, any signal which is totally predictable carries no information. In the case of the sine wave this is clear because it represents a single frequency and so has no bandwidth.

At the opposite extreme a signal such as noise is completely unpredictable and as a result all codecs find noise *difficult*. The most efficient way of coding noise is PCM. A codec which is designed using the statistics of real material should not be tested with random noise because it is not a representative test. Second, a codec which performs well with clean source material may perform badly with source material containing superimposed noise. Most practical compression units require some form of pre-processing before the compression stage proper and appropriate noise reduction should be incorporated into the pre-processing if noisy signals are anticipated. It will also be necessary to restrict the degree of compression applied to noisy signals.

All real signals fall part-way between the extremes of total predictability and total unpredictability or noisiness. If the bandwidth (set by the

Google Exhibit No. 1005
Page 017

Google Exhibit No. 1063
Page 0272

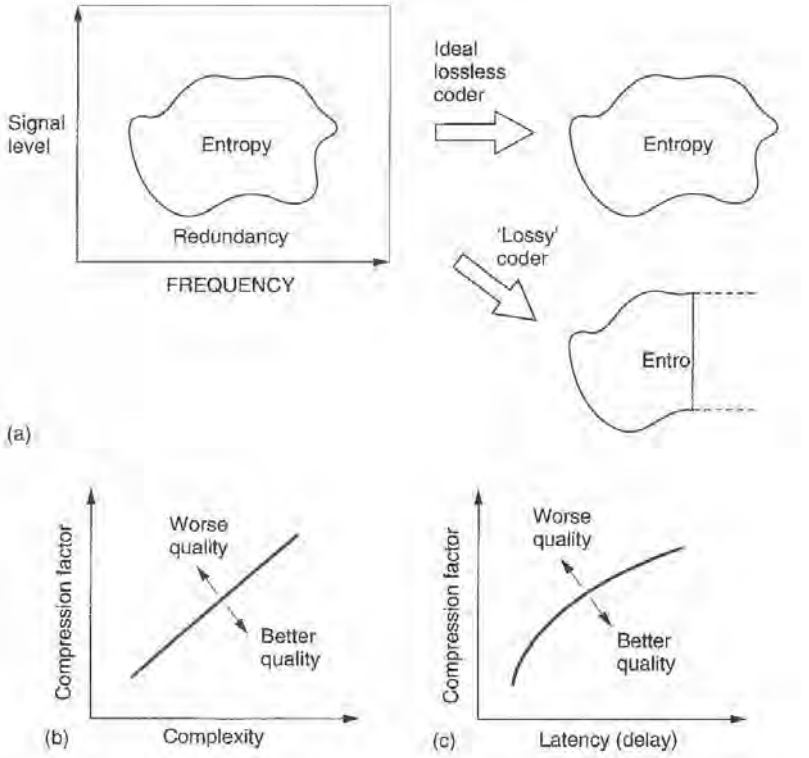


Figure 1.5 (a) A perfect coder removes only the redundancy from the input signal and results in subjectively lossless coding. If the remaining entropy is beyond the capacity of the channel some of it must be lost and the codec will then be lossy. An imperfect coder will also be lossy as it fails to keep all entropy. (b) As the compression factor rises, the complexity must also rise to maintain quality. (c) High compression factors also tend to increase latency or delay through the system.

sampling rate) and the dynamic range (set by the wordlength) of the transmission system are used to delineate an area, this sets a limit on the information capacity of the system. Figure 1.5(a) shows that most real signals only occupy part of that area. The signal may not contain all frequencies, or it may not have full dynamics at certain frequencies.

Entropy can be thought of as a measure of the actual area occupied by the signal. This is the area that *must* be transmitted if there are to be no subjective differences or *artifacts* in the received signal. The remaining area is called the *redundancy* because it adds nothing to the information conveyed. Thus an ideal coder could be imagined which miraculously sorts out the entropy from the redundancy and only sends the former. An ideal decoder would then re-create the original impression of the information quite perfectly. As the ideal is approached, the coder complexity and the latency or delay both rise. Figure 1.5(b) shows how

Google Exhibit No. 1005
Page 018

Google Exhibit No. 1063
Page 0273

complexity increases with compression factor. The additional complexity of MPEG-4 over MPEG-2 is obvious from this. Figure 1.5(c) shows how increasing the codec latency can improve the compression factor.

Obviously we would have to provide a channel which could accept whatever entropy the coder extracts in order to have transparent quality. As a result moderate coding gains which only remove redundancy need not cause artifacts and result in systems which are described as *subjectively lossless*. If the channel capacity is not sufficient for that, then the coder will have to discard some of the entropy and with it useful information. Larger coding gains which remove some of the entropy must result in artifacts. It will also be seen from Figure 1.5 that an imperfect coder will fail to separate the redundancy and may discard entropy instead, resulting in artifacts at a sub-optimal compression factor.

A single variable-rate transmission or recording channel is traditionally unpopular with channel providers, although newer systems such as ATM support variable rate. Digital transmitters used in DVB have a fixed bit rate. The variable rate requirement can be overcome by combining several compressed channels into one constant rate transmission in a way which flexibly allocates data rate between the channels. Provided the material is unrelated, the probability of all channels reaching peak entropy at once is very small and so those channels which are at one instant passing easy material will make available transmission capacity for those channels which are handling difficult material. This is the principle of statistical multiplexing.

Where the same type of source material is used consistently, e.g. English text, then it is possible to perform a statistical analysis on the frequency with which particular letters are used. Variable-length coding is used in which frequently used letters are allocated short codes and letters which occur infrequently are allocated long codes. This results in a lossless code. The well-known Morse code used for telegraphy is an example of this approach. The letter e is the most frequent in English and is sent with a single dot. An infrequent letter such as z is allocated a long complex pattern. It should be clear that codes of this kind which rely on a prior knowledge of the statistics of the signal are only effective with signals actually having those statistics. If Morse code is used with another language, the transmission becomes significantly less efficient because the statistics are quite different; the letter z, for example, is quite common in Czech.

The Huffman code[5] is also one which is designed for use with a data source having known statistics. The probability of the different code values to be transmitted is studied, and the most frequent codes are arranged to be transmitted with short wordlength symbols. As the probability of a code value falls, it will be allocated longer wordlength.

Google Exhibit No. 1005
Page 019

Google Exhibit No. 1063
Page 0274

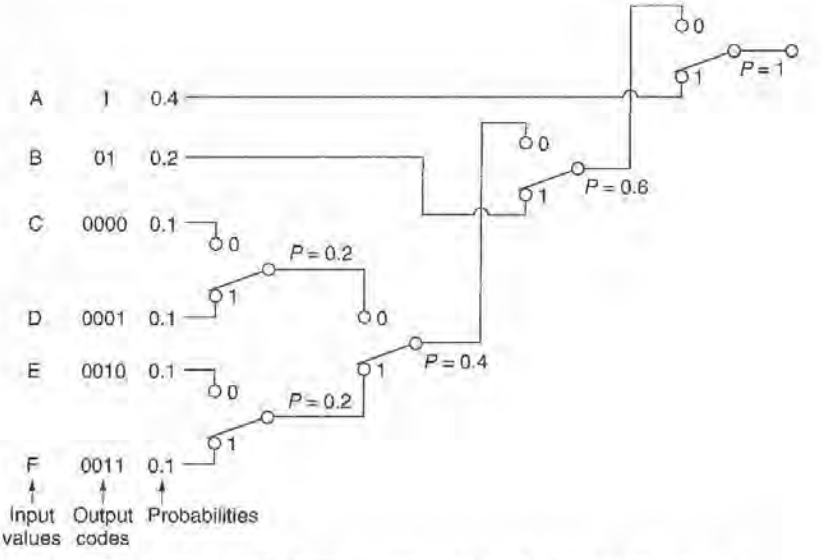


Figure 1.6 The Huffman code achieves compression by allocating short codes to frequent values. To aid deserializing the short codes are not prefixes of longer codes.

The Huffman code is used in conjunction with a number of compression techniques and is shown in Figure 1.6.

The input or *source* codes are assembled in order of descending probability. The two lowest probabilities are distinguished by a single code bit and their probabilities are combined. The process of combining probabilities is continued until unity is reached and at each stage a bit is used to distinguish the path. The bit will be a zero for the most probable path and one for the least. The compressed output is obtained by reading the bits which describe which path to take going from right to left.

In the case of computer data, there is no control over the data statistics. Data to be recorded could be instructions, images, tables, text files and so on; each having their own code value distributions. In this case a coder relying on fixed source statistics will be completely inadequate. Instead a system is used which can learn the statistics as it goes along. The Lempel–Ziv–Welch (LZW) lossless codes are in this category. These codes build up a conversion table between frequent long source data strings and short transmitted data codes at both coder and decoder and initially their compression factor is below unity as the contents of the conversion tables are transmitted along with the data. However, once the tables are established, the coding gain more than compensates for the initial loss. In some applications, a continuous analysis of the frequency of code selection is made and if a data string in the table is no longer being used with sufficient frequency it can be deselected and a more common string substituted.

Google Exhibit No. 1005
Page 020

Google Exhibit No. 1063
Page 0275

Lossless codes are less common for audio and video coding where perceptive codes are permissible. The perceptive codes often obtain a coding gain by shortening the wordlength of the data representing the signal waveform. This must increase the noise level and the trick is to ensure that the resultant noise is placed at frequencies where human senses are least able to perceive it. As a result although the received signal is measurably different from the source data, it can *appear* the same to the human listener or viewer at moderate compressions factors. As these codes rely on the characteristics of human sight and hearing, they can only be fully tested subjectively.

The compression factor of such codes can be set at will by choosing the wordlength of the compressed data. Whilst mild compression will be undetectable, with greater compression factors, artifacts become noticeable. Figure 1.5 shows that this is inevitable from entropy considerations.

1.7 Video compression

Video signals exist in four dimensions: these are the attributes of the pixel, the horizontal and vertical spatial axes and the time axis. Compression can be applied in any or all of those four dimensions. MPEG assumes an eight-bit colour difference signal as the input, requiring rounding if the source is ten-bit. The sampling rate of the colour signals is less than that of the luminance. This is done by downsampling the colour samples horizontally and generally vertically as well. Essentially an MPEG system has three parallel simultaneous channels, one for luminance and two colour difference, which after coding are multiplexed into a single bitstream.

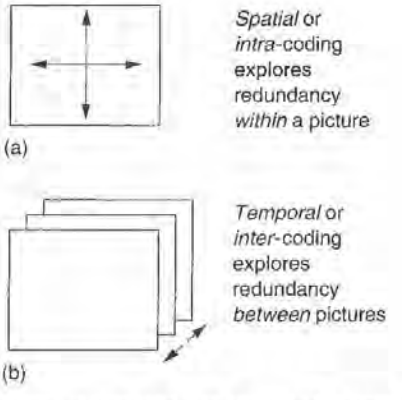


Figure 1.7 (a) Spatial or intra-coding works on individual images. (b) Temporal or inter-coding works on successive images.

Google Exhibit No. 1005
Page 021

Google Exhibit No. 1063
Page 0276

Figure 1.7(a) shows that when individual pictures are compressed without reference to any other pictures, the time axis does not enter the process which is therefore described as *intra-coded* (intra = within) compression. The term *spatial coding* will also be found. It is an advantage of intra-coded video that there is no restriction to the editing which can be carried out on the picture sequence. As a result compressed VTRs such as Digital Betacam, DVC and D-9 use spatial coding. Cut editing may take place on the compressed data directly if necessary. As spatial coding treats each picture independently, it can employ certain techniques developed for the compression of still pictures. The ISO JPEG (Joint Photographic Experts Group) compression standards[6,7] are in this category. Where a succession of JPEG coded images are used for television, the term 'Motion JPEG' will be found.

Greater compression factors can be obtained by taking account of the redundancy from one picture to the next. This involves the time axis, as Figure 1.7(b) shows, and the process is known as *inter-coded* (inter = between) or *temporal* compression.

Temporal coding allows a higher compression factor, but has the disadvantage that an individual picture may exist only in terms of the differences from a previous picture. Clearly editing must be undertaken with caution and arbitrary cuts simply cannot be performed on the MPEG bitstream. If a previous picture is removed by an edit, the difference data will then be insufficient to re-create the current picture.

1.7.1 Intra-coded compression

Intra-coding works in three dimensions on the horizontal and vertical spatial axes and on the sample values. Analysis of typical television pictures reveals that whilst there is a high spatial frequency content due to detailed areas of the picture, there is a relatively small amount of energy at such frequencies. Often pictures contain sizeable areas in which the same or similar pixel values exist. This gives rise to low spatial frequencies. The average brightness of the picture results in a substantial zero frequency component. Simply omitting the high-frequency components is unacceptable as this causes an obvious softening of the picture.

A coding gain can be obtained by taking advantage of the fact that the amplitude of the spatial components falls with frequency. It is also possible to take advantage of the eye's reduced sensitivity to noise in high spatial frequencies. If the spatial frequency spectrum is divided into frequency bands the high-frequency bands can be described by fewer bits not only because their amplitudes are smaller but also because more noise can be tolerated. The wavelet transform (MPEG-4 only) and the discrete cosine transform used in JPEG and MPEG-1, MPEG-2 and MPEG-4 allow

Google Exhibit No. 1005
Page 022

Google Exhibit No. 1063
Page 0277

two-dimensional pictures to be described in the frequency domain and these are discussed in Chapter 3.

1.7.2 Inter-coded compression

Inter-coding takes further advantage of the similarities between successive pictures in real material. Instead of sending information for each picture separately, inter-coders will send the difference between the previous picture and the current picture in a form of differential coding. Figure 1.8 shows the principle. A picture store is required at the coder to allow comparison to be made between successive pictures and a similar store is required at the decoder to make the previous picture available. The difference data may be treated as a picture itself and subjected to some form of transform-based spatial compression.

The simple system of Figure 1.8(a) is of limited use as in the case of a transmission error, every subsequent picture would be affected. Channel switching in a television set would also be impossible. In practical systems a modification is required. One approach is the so-called 'leaky predictor' in which the next picture is predicted from a limited number of previous pictures rather than from an indefinite number. As a result errors cannot propagate indefinitely. The approach used in MPEG is that periodically some absolute picture data are transmitted in place of difference data.

Figure 1.8(b) shows that absolute picture data, known as *I* or *intra pictures* are interleaved with pictures which are created using difference data, known as *P* or *predicted* pictures. The *I* pictures require a large amount of data, whereas the *P* pictures require fewer data. As a result the instantaneous data rate varies dramatically and buffering has to be used to allow a constant transmission rate. The leaky predictor needs less buffering as the compression factor does not change so much from picture to picture.

The *I* picture and all of the *P* pictures prior to the next *I* picture are called a group of pictures (GOP). For a high compression factor, a large number of *P* pictures should be present between *I* pictures, making a long GOP. However, a long GOP delays recovery from a transmission error. The compressed bitstream can only be edited at *I* pictures as shown.

In the case of moving objects, although their appearance may not change greatly from picture to picture, the data representing them on a fixed sampling grid will change and so large differences will be generated between successive pictures. It is a great advantage if the effect of motion can be removed from difference data so that they only reflect the changes in appearance of a moving object since a much greater coding gain can then be obtained. This is the objective of motion compensation.

Google Exhibit No. 1005
Page 023

Google Exhibit No. 1063
Page 0278

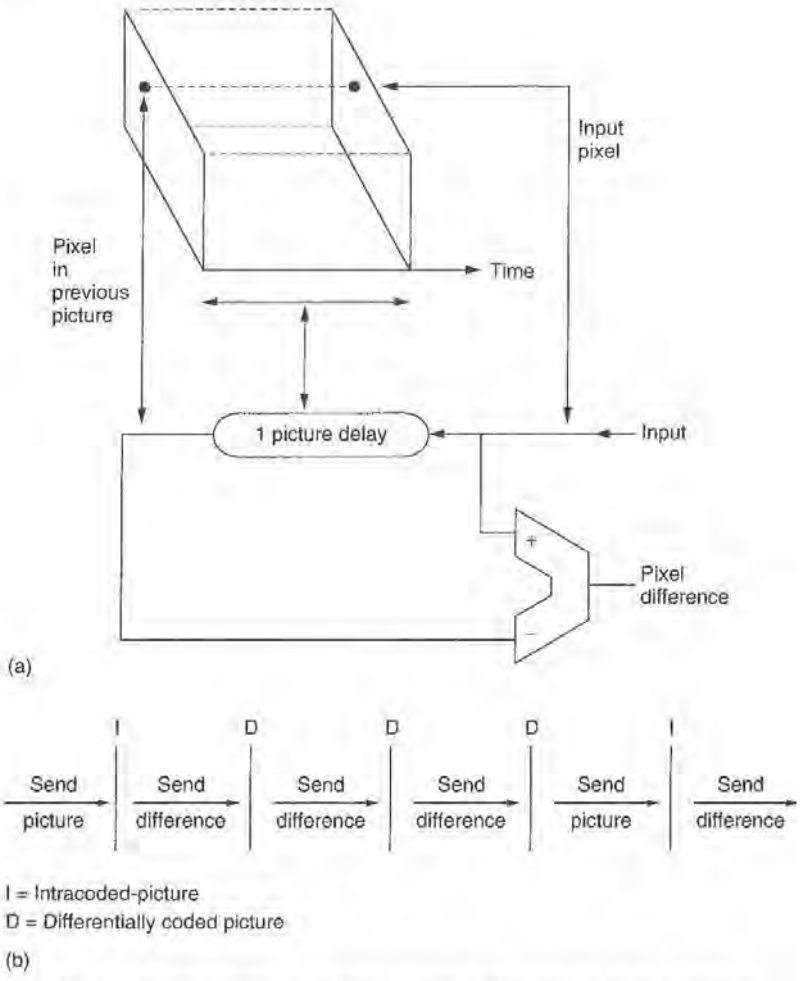


Figure 1.8 An inter-coded system (a) uses a delay to calculate the pixel differences between successive pictures. To prevent error propagation, intra-coded pictures (b) may be used periodically.

1.7.3 Introduction to motion compensation

In real television program material objects move around before a fixed camera or the camera itself moves. Motion compensation is a process which effectively measures motion of objects from one picture to the next so that it can allow for that motion when looking for redundancy between pictures. Figure 1.9 shows that moving pictures can be expressed in a three-dimensional space which results from the screen area moving along the time axis. In the case of still objects, the only motion is along the time

Google Exhibit No. 1005
Page 024

Google Exhibit No. 1063
Page 0279

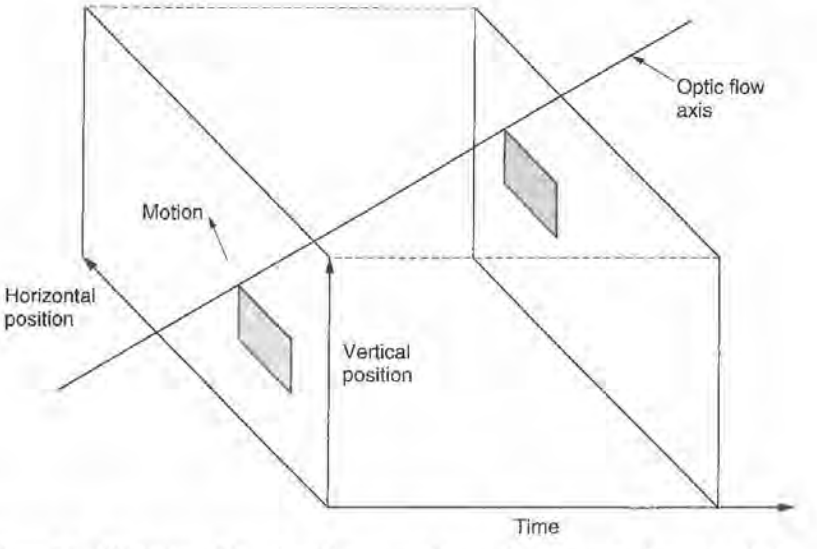


Figure 1.9 Objects travel in a three-dimensional space along the optic flow axis which is only parallel to the time axis if there is no movement.

axis. However, when an object moves, it does so along the *optic flow axis* which is not parallel to the time axis. The optic flow axis is the locus of a point on a moving object as it takes on various screen positions.

It will be clear that the data values representing a moving object change with respect to the time axis. However, looking along the optic flow axis the appearance of an object only changes if it deforms, moves into shadow or rotates. For simple translational motions the data representing an object are highly redundant with respect to the optic flow axis. Thus if the optic flow axis can be located, coding gain can be obtained in the presence of motion.

A motion-compensated coder works as follows. A reference picture is sent, but is also locally stored so that it can be compared with another picture to find motion vectors for various areas of the picture. The reference picture is then shifted according to these vectors to cancel inter-picture motion. The resultant *predicted* picture is compared with the actual picture to produce a *prediction error* also called a *residual*. The prediction error is transmitted with the motion vectors. At the receiver the reference picture is also held in a memory. It is shifted according to the transmitted motion vectors to re-create the predicted picture and then the prediction error is added to it to re-create the original.

In prior compression schemes the predicted picture followed the reference picture. In MPEG this is not the case. Information may be brought back from a later picture or forward from an earlier picture as appropriate.

Google Exhibit No. 1005
Page 025

Google Exhibit No. 1063
Page 0280

1.7.4 Film-originated video compression

Film can be used as the source of video signals if a telecine machine is used. The most common frame rate for film is 24 Hz, whereas the field rates of television are 50 Hz and 60 Hz. This incompatibility is patched over in two different ways. In 50 Hz telecine, the film is simply played slightly too fast so that the frame rate becomes 25 Hz. Then each frame is converted into two television fields giving the correct 50 Hz field rate. In 60 Hz telecine, the film travels at the correct speed, but alternate frames are used to produce two fields then three fields. The technique is known as 3:2 pulldown. In this way two frames produce five fields and so the correct 60 Hz field rate results. The motion portrayal of telecine is not very good as moving objects judder, especially in 60 Hz systems. Figure 1.10 shows how the optic flow is portrayed in film-originated video.

When film-originated video is input to a compression system, the disturbed optic flow will play havoc with the motion-compensation system. In a 50 Hz system there appears to be no motion between the two fields which have originated from the same film frame, whereas between the next two fields large motions will exist. In 60 Hz systems, the motion will be zero for three fields out of five.

With such inputs, it is more efficient to adopt a different processing mode which is based upon the characteristics of the original film. Instead of attempting to manipulate fields of video, the system de-interlaces pairs of fields in order to reconstruct the original film frames. This can be done by a fairly simple motion detector. When substantial motion is measured

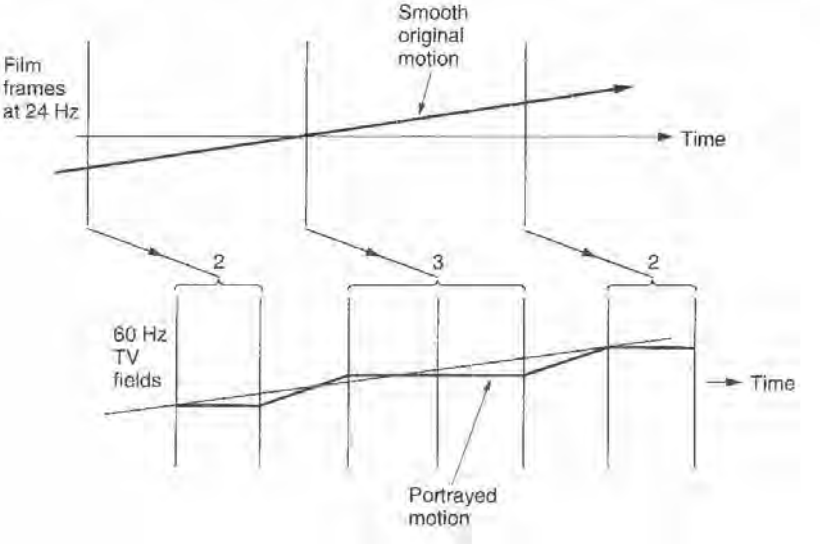


Figure 1.10 Telecine machines must use 3:2 pulldown to produce 60 Hz field rate video

Google Exhibit No. 1005
Page 026

Google Exhibit No. 1063
Page 0281

between successive fields in the output of a telecine, this is taken to mean that the fields have come from different film frames. When negligible motion is detected between fields, this is taken to indicate that the fields have come from the same film frame.

In 50 Hz video it is quite simple to find the sequence and produce de-interlaced frames at 25 Hz. In 60 Hz 3:2 pulldown video the problem is slightly more complex because it is necessary to locate the frames in which three fields are output so that the third field can be discarded, leaving, once more, de-interlaced frames at 25 Hz. Whilst it is relatively straightforward to lock-on to the 3:2 sequence with direct telecine output signals, if the telecine material has been edited on videotape the 3:2 sequence may contain discontinuities. In this case it is necessary to provide a number of field stores in the de-interlace unit so that a series of fields can be examined to locate the edits. Once telecine video has been de-interlaced back to frames, intra- and inter-coded compression can be employed using frame-based motion compensation.

MPEG transmissions include flags which tell the decoder the origin of the material. Material originating at 24 Hz but converted to interlaced video does not have the motion attributes of interlace because the lines in two fields have come from the same point on the time axis. Two fields can be combined to create a progressively scanned frame. In the case of 3:2 pulldown material, the third field need not be sent at all as the decoder can easily repeat a field from memory. As a result the same compressed film material can be output at 50 or 60 Hz as required.

Recently conventional telecine machines have been superseded by the *datacine* which scans each film frame into a pixel array which can be made directly available to the MPEG encoder without passing through an intermediate digital video standard. Datacines are used extensively for mastering DVDs from film stock.

1.8 Introduction to MPEG-1

As mentioned above, the intention of MPEG-1 is to deliver video and audio at the same bit rate as a conventional audio CD. As the bit rate was a given, this was achieved by subsampling to half the definition of conventional television. In order to have a constant input bit rate irrespective of the frame rate, 25 Hz systems have a picture size of 352 × 288 pixels whereas 30 Hz systems have a picture size of 352 × 240 pixels. This is known as *common intermediate format* (CIF). If the input is conventional interlaced video, CIF can be obtained by discarding alternate fields and downsampling the remaining active lines by a factor of two.

Google Exhibit No. 1005
Page 027

Google Exhibit No. 1063
Page 0282

As interlaced systems have very poor vertical resolution, down-sampling to CIF actually does little damage to still images, although the very low picture rates damage motion portrayal.

Although MPEG-1 appeared rather rough on screen, this was due to the very low bit rate. It is more important to appreciate that MPEG-1 introduced the great majority of the coding tools which would continue to be used in MPEG-2 and MPEG-4. These included an elementary stream syntax, bidirectional motion-compensated coding, buffering and rate control. Many of the spatial coding principles of MPEG-1 were taken from JPEG. MPEG-1 also specified audio compression of up to two channels.

1.9 MPEG-2: Profiles and Levels

MPEG-2 builds upon MPEG-1 by adding interlace capability as well as a greatly expanded range of picture sizes and bit rates. The use of scaleable systems is also addressed, along with definitions of how multiple MPEG bitstreams can be multiplexed. As MPEG-2 is an extension of MPEG-1, it is easy for MPEG-2 decoders to handle MPEG-1 data. In a sense an MPEG-1 bitstream is an MPEG-2 bitstream which has a restricted vocabulary and so can be readily understood by an MPEG-2 decoder.

MPEG-2 has too many applications to solve with a single standard and so it is subdivided into Profiles and Levels. Put simply a Profile describes a degree of complexity whereas a Level describes the picture size or resolution which goes with that Profile. Not all Levels are supported at all Profiles. Figure 1.11 shows the available combinations. In principle there

Levels \ Profiles	Simple	Main	4:2:2	SNR	Spatial	High
High		4:2:0 1920 × 1152 90 Mb/S				4:2:0 or 4:2:2 1920 × 1152 100 Mb/S
High 1440		4:2:0 1440 × 1152 60 Mb/S			4:2:0 1440 × 1152 60 Mb/S	4:2:0 or 4:2:2 1440 × 1152 80 Mb/S
Main	4:2:0 720 × 576 15 Mb/S NO B	4:2:0 720 × 576 15 Mb/S	4:2:2 720 × 608 50 Mb/S	4:2:0 720 × 576 15 Mb/S		4:2:0 or 4:2:2 720 × 576 20 Mb/S
Low		4:2:0 352 × 288 4 Mb/S		4:2:0 352 × 288 4 Mb/S		

Figure 1.11 Profiles and Levels in MPEG-2. See text for details.

Google Exhibit No. 1005
Page 028

Google Exhibit No. 1063
Page 0283

are twenty-four of these, but not all have been defined. An MPEG-2 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 television). The High-1440 Level is a high-definition scheme which doubles the definition compared to 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.

Figure 1.12(a) shows that in a conventional MPEG coder, by heavily requantizing coefficients a picture with moderate signal-to-noise ratio

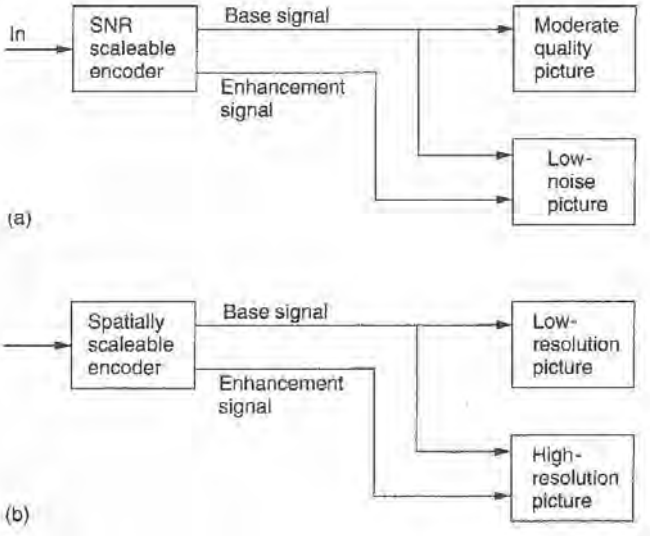


Figure 1.12 (a) An SNR scaleable encoder produces a 'noisy' signal and a noise cancelling signal. (b) A spatially scaleable encoder produces a low-resolution picture and a resolution-enhancing picture.

Google Exhibit No. 1005
Page 029

Google Exhibit No. 1063
Page 0284

results. If, however, that picture is locally decoded and subtracted pixel by pixel from the original, a 'quantizing noise' picture would result. This can be compressed and transmitted as the helper signal. A simple decoder only decodes the main 'noisy' bitstream, but a more complex decoder can decode both bitstreams and combine them to produce a low-noise picture. This is the principle of SNR scaleability.

As an alternative, Figure 1.12(b) shows that by coding only the lower spatial frequencies in a HDTV picture a base bitstream can be made which 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 can be coded into a helper signal. A suitable decoder could combine the main and helper signals to re-create the HDTV picture. This is the principle of spatial scaleability.

The High Profile supports both SNR and spatial scaleability as well as allowing the option of 4:2:2 sampling (see Section 2.11).

The 4:2:2 Profile has been developed for improved compatibility with existing digital television production equipment. This allows 4:2:2 working without requiring the additional complexity of using the High Profile. For example a HP ML decoder must support SNR scaleability which is not a requirement for production.

MPEG-2 increased the number of audio channels possible to five whilst remaining compatible with MPEG-1 audio. MPEG-2 subsequently introduced a more efficient audio coding scheme known as MPEG-2 AAC (advanced audio coding) which is not backwards compatible with the earlier audio coding schemes.

1.10 Introduction to MPEG-4

MPEG-4 introduces a number of new coding tools as shown in Figure 1.13. In MPEG-1 and MPEG-2 the motion compensation is based on regular fixed-size areas of image known as *macroblocks*. Whilst this works well at the designed bit rates, there will always be some inefficiency due to real moving objects failing to align with macroblock boundaries. This will increase the residual bit rate. In MPEG-4, moving objects can be coded as arbitrary shapes. Figure 1.14 shows that a background can be coded quite independently from objects in front of it. Object motion can then be described with vectors and much-reduced residual data. According to the Profile, objects may be two-dimensional, three-dimensional and opaque or translucent. The decoder must contain effectively a layering vision mixer which is capable of prioritizing image data as a function of how close it is to the viewer. The picture coding of MPEG-4 is known as texture coding and is more advanced than the

Google Exhibit No. 1005
Page 030

Google Exhibit No. 1063
Page 0285

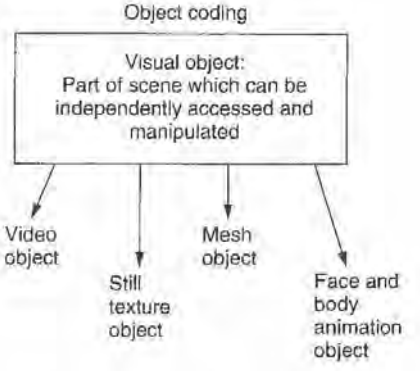


Figure 1.13 MPEG-4 introduces a number of new coding tools over those of earlier MPEG standards. These include object coding, mesh coding, still picture coding and face and body animation.

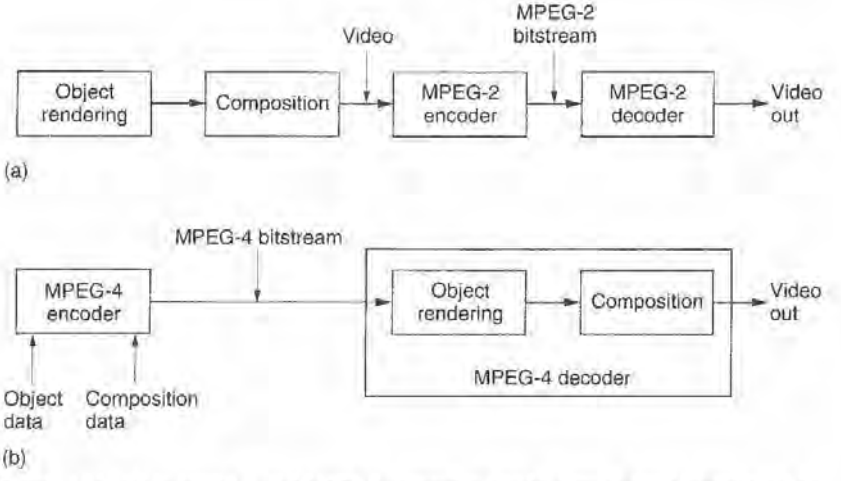


Figure 1.14 (a) In MPEG-1 and MPEG-2, computer graphic images must be rendered to video before coding. (b) In contrast, MPEG-4 may move the rendering process to the decoder, reducing the bit rate needed with the penalty of increased decoder complexity.

MPEG-2 equivalent, using more lossless predictive coding for pixel values, coefficients and vectors.

In addition to motion compensation, MPEG-4 can describe how an object changes its perspective as it moves using a technique called mesh coding. By warping another image, the prediction of the present image is improved. MPEG-4 also introduces coding for still images using DCT or wavelets.

Although MPEG-2 supported some scaleability, MPEG-4 also takes this further. In addition to spatial and noise scaleability, MPEG-4 also allows

Google Exhibit No. 1005
Page 031

Google Exhibit No. 1063
Page 0286

temporal scaleability where a base level bitstream having a certain frame rate may be augmented by an additional enhancement bitstream to produce a decoder output at a higher frame rate. This is important as it allows a way forward from the marginal frame rates of today's film and television formats whilst remaining backwards compatible with traditional equipment. The comprehensive scaleability of MPEG-4 is equally important in networks where it allows the user the best picture possible for the available bit rate.

MPEG-4 also introduces standards for face and body animation. Specialized vectors allow a still picture of a face and optionally a body to be animated to allow expressions and gestures to accompany speech at very low bit rates. In some senses MPEG-4 has gone upstream of the video signal which forms the input to MPEG-1 and MPEG-2 coders to analyse ways in which the video signal was rendered. Figure 1.14(a) shows that in a system using MPEG-1 and MPEG-2, all rendering and production steps take place before the encoder. Figure 1.14(b) shows that in MPEG-4, some of these steps can take place in the decoder. The advantage is that fewer data need to be transmitted. Some of these data will be rendering instructions which can be very efficient and result in a high compression factor. As a significant part of the rendering takes place in the decoder, computer graphics generators can be designed directly to output an MPEG-4 bitstream. In interactive systems such as simulators and video games, inputs from the user can move objects around the screen. The disadvantage is increased decoder complexity, but as the economics of digital processing continues to advance this is hardly a serious concern.

As might be expected, the huge range of coding tools in MPEG-4 is excessive for many applications. As with MPEG-2 this has been dealt with using Profiles and Levels. Figure 1.15 shows the range of Visual

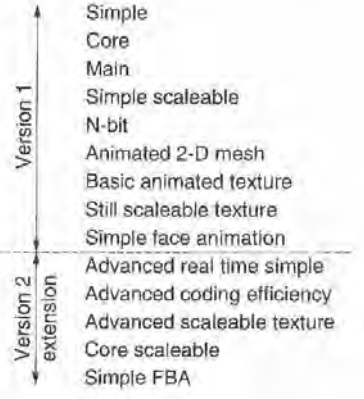


Figure 1.15 The visual object types supported by MPEG-4 in versions 1 and 2.

Google Exhibit No. 1005
Page 032

Google Exhibit No. 1063
Page 0287

Version 1 visual profiles

	Object types / Profiles	Simple	Core	Main	Simple scaleable	N-bit	Animated 2D mesh	Basic animated texture	Scaleable texture	Simple face
1.	Simple	×								
2.	Simple scaleable	×			×					
3.	Core	×	×							
4.	Main	×	×	×					×	
5.	N-bit	×	×			×				
6.	Hybrid	×	×				×	×	×	×
7.	Basic animated texture							×	×	×
8.	Scaleable texture								×	
9.	Simple FA									×

Version 2 visual profiles

	Object types / Profiles	Simple	Simple scalable	Core	Core scalable	Advanced real time simple	Advanced coding efficiency	Advanced scalable texture	Simple FBA
V2-1.	Advanced real time simple	×				×			
V2-2.	Core scaleable	×	×	×	×				
V2-3.	Advanced coding efficiency	×		×			×		
V2-4.	Advanced core	×		×				×	
V2-5.	Advanced scalable texture							×	
V2-6.	Simple FBA								×

Figure 1.16 The visual object types supported by each visual profile of MPEG-4.

Object types in version 1 of MPEG-4 and as expanded in version 2. For each visual object type the coding tools needed is shown. Figure 1.16 shows the relationship between the Visual Profiles and the Visual Object types supported by each Profile. The crossover between computer-generated and natural images is evident in the Profile structure where Profiles 1–5 cover natural images, Profiles 8 and 9 cover rendered images and Profiles 6 and 7 cover hybrid natural/rendered images. It is only possible to give an introduction here and more detail is provided in Chapter 5. MPEG-4 also extends the boundaries of audio coding. The MPEG-2 AAC technique is extended in MPEG-4 by some additional tools. New tools are added which allow operation at very low bit rates for speech applications. Also introduced is the concept of *structured audio* in which the audio waveform is synthesized at the decoder from a bitstream which is essentially a digital musical score.

Google Exhibit No. 1005
Page 033

Google Exhibit No. 1063
Page 0288

1.11 Audio compression

Perceptive coding in audio relies on the principle of auditory masking, which is treated in detail in section 4.1. Masking causes the ear/brain combination to be less sensitive to sound at one frequency in the presence of another at a nearby frequency. If a first tone is present in the input, then it will mask signals of lower level at nearby frequencies. The quantizing of the first tone and of further tones at those frequencies can be made coarser. Fewer bits are needed and a coding gain results. The increased quantizing error is allowable if it is masked by the presence of the first tone.

1.11.1 Sub-band coding

Sub-band coding mimics the frequency analysis mechanism of the ear and splits the audio spectrum into a large number of different bands. Signals in these bands can then be quantized independently. The quantizing error which results is confined to the frequency limits of the band and so it can be arranged to be masked by the program material. The techniques used in Layers I and II of MPEG audio are based on sub-band coding as are those used in DCC (Digital Compact Cassette).

1.11.2 Transform coding

In transform coding the time-domain audio waveform is converted into a frequency domain representation such as a Fourier, discrete cosine or wavelet transform (see Chapter 3). Transform coding takes advantage of the fact that the amplitude or envelope of an audio signal changes relatively slowly and so the coefficients of the transform can be transmitted relatively infrequently. Clearly such an approach breaks down in the presence of transients and adaptive systems are required in practice. Transients cause the coefficients to be updated frequently whereas in stationary parts of the signal such as sustained notes the update rate can be reduced. Discrete cosine transform (DCT) coding is used in Layer III of MPEG audio and in the compression system of the Sony MiniDisc.

1.11.3 Predictive coding

In a predictive coder there are two identical predictors, one in the coder and one in the decoder. Their job is to examine a run of previous data

Google Exhibit No. 1005
Page 034

Google Exhibit No. 1063
Page 0289

values and to extrapolate forward to estimate or predict what the next value will be. This is subtracted from the *actual* next code value at the encoder to produce a prediction error which is transmitted. The decoder then adds the prediction error to its own prediction to obtain the output code value again.

Prediction can be used in the time domain, where sample values are predicted, or in the frequency domain where coefficient values are predicted. Time-domain predictive coders work with a short encode and decode delay and are useful in telephony where a long loop delay causes problems. Frequency prediction is used in AC-3 and MPEG AAC.

1.12 MPEG bitstreams

MPEG supports a variety of bitstream types for various purposes and these are shown in Figure 1.17. The output of a single compressor (video or audio) is known as an elementary stream. In transmission, many elementary streams will be combined to make a transport stream. Multiplexing requires blocks or packets of constant size. It is advantageous if these are short so that each elementary stream in the multiplex can receive regular data. A transport stream has a complex structure because it needs to incorporate metadata indicating which audio

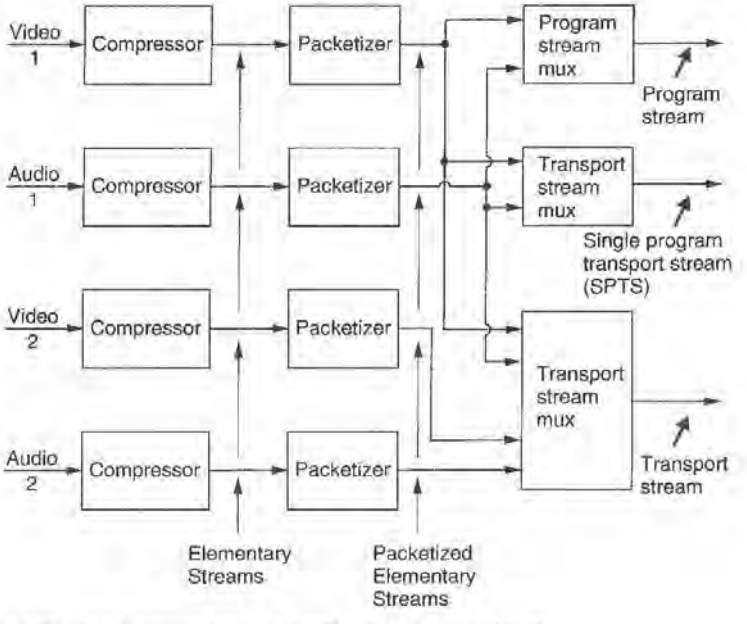


Figure 1.17 The bitstream types of MPEG-2. See text for details.

Google Exhibit No. 1005
Page 035

Google Exhibit No. 1063
Page 0290

elementary streams and ancillary data are associated with which video elementary stream. It is possible to have a single program transport stream (SPTS) which carries only the elementary streams of one TV program.

For certain purposes, such as recording a single elementary stream, the transport stream is not appropriate. The small packets of the transport stream each require a header and this wastes storage space. In this case a program stream can be used. A program stream is a simplified bitstream which multiplexes audio and video for a single program together, provided they have been encoded from a common locked clock. Unlike a transport stream, the blocks are larger and are not necessarily of fixed size.

1.13 Drawbacks of compression

By definition, compression removes redundancy from signals. Redundancy is, however, essential to making data resistant to errors. As a result, compressed data are more sensitive to errors than uncompressed data. Thus transmission systems using compressed data must incorporate more powerful error-correction strategies and avoid compression techniques which are notoriously sensitive. As an example, the Digital Betacam format uses relatively mild compression and yet requires 20 per cent redundancy whereas the D-5 format does not use compression and only requires 17 per cent redundancy even though it has a recording density 30 per cent higher. Techniques using tables such as the Lempel–Ziv–Welch codes are very sensitive to bit errors as an error in the transmission of a table value results in bit errors every time that table location is accessed. This is known as error propagation. Variable-length techniques such as the Huffman code are also sensitive to bit errors. As there is no fixed symbol size, the only way the decoder can parse a serial bitstream into symbols is to increase the assumed wordlength a bit at a time until a code value is recognized. The next bit must then be the first bit in the next symbol. A single bit in error could cause the length of a code to be wrongly assessed and then all subsequent codes would also be wrongly decoded until synchronization could be re-established. Later variable-length codes sacrifice some compression efficiency in order to offer better resynchronization properties.

In non-real-time systems such as computers an uncorrectable error results in reference to the back-up media. In real-time systems such as audio and video this is impossible and concealment must be used. However, concealment relies on redundancy and compression reduces the degree of redundancy. Media such as hard disks can be verified so that uncorrectable errors are virtually eliminated, but tape is prone to

Google Exhibit No. 1005
Page 036

Google Exhibit No. 1063
Page 0291

dropouts which will exceed the burst-correcting power of the replay system from time to time. For this reason the compression factors used on audio or video tape should be moderate.

As perceptive coders introduce noise, it will be clear that in a concatenated system the second codec could be confused by the noise due to the first. If the codecs are identical then each may well make, or better still be designed to make, the same decisions when they are in tandem. If the codecs are not identical the results could be disappointing. Signal manipulation between codecs can also result in artifacts which were previously undetectable becoming visible because the signal which was masking them is no longer present.

In general, compression should not be used for its own sake, but only where a genuine bandwidth or cost bottleneck exists. Even then the mildest compression possible should be used. Whilst high compression factors are permissible for final delivery of material to the consumer, they are not advisable prior to any post-production stages. For contribution material, lower compression factors are essential and this is sometimes referred to as *mezzanine* level compression.

One practical drawback of compression systems is that they are largely generic in structure and the same hardware can be operated at a variety of compression factors. Clearly the higher the compression factor, the cheaper the system will be to operate so there will be economic pressure to use high compression factors. Naturally the risk of artifacts is increased and so there is (or should be) counterpressure from those with engineering skills to moderate the compression. The way of the world at the time of writing is that the accountants have the upper hand. This was not a problem when there were fixed standards such as PAL and NTSC, as there was no alternative but to adhere to them. Today there is plenty of evidence that the variable compression factor control is being turned too far in the direction of economy.

It has been seen above that concatenation of compression systems should be avoided as this causes generation loss. Generation loss is worse if the codecs are different. Interlace is a legacy compression technique and if concatenated with MPEG, generation loss will be exaggerated. In theory and in practice better results are obtained in MPEG for the same bit rate if the input is progressively scanned. Consequently the use of interlace with MPEG coders cannot be recommended for new systems. Chapter 5 explores this theme in greater detail.

1.14 Compression pre-processing

Compression relies completely on identifying redundancy in the source material. Consequently anything which reduces that redundancy will

Google Exhibit No. 1005
Page 037

Google Exhibit No. 1063
Page 0292

have a damaging effect. Noise is particularly undesirable as it creates additional spatial frequencies in individual pictures as well as spurious differences between pictures. Where noisy source material is anticipated some form of noise reduction will be essential.

When high compression factors must be used to achieve a low bit rate, it is inevitable that the level of artifacts will rise. In order to contain the artifact level, it is necessary to restrict the source entropy prior to the coder. This may be done by spatial low-pass filtering to reduce the picture resolution, and may be combined with downsampling to reduce the number of pixels per picture. In some cases, such as teleconferencing, it will also be necessary to reduce the picture rate. At very low bit rates the use of interlace becomes acceptable as a pre-processing stage providing downsampling prior to the MPEG compression.

A compression pre-processor will combine various types of noise reduction (see Chapter 3) with spatial and temporal downsampling.

1.15 Some guidelines

Although compression techniques themselves are complex, there are some simple rules which can be used to avoid disappointment. Used wisely, MPEG compression has a number of advantages. Used in an inappropriate manner, disappointment is almost inevitable and the technology could get a bad name. The next few points are worth remembering.

- Compression technology may be exciting, but if it is not necessary it should not be used.
- If compression is to be used, the degree of compression should be as small as possible; i.e. use the highest practical bit rate.
- Cascaded compression systems cause loss of quality and the lower the bit rates, the worse this gets. Quality loss increases if any post-production steps are performed between compressions.
- Avoid using interlaced video with MPEG.
- Compression systems cause delay.
- Compression systems work best with clean source material. Noisy signals or poorly decoded composite video give poor results.
- Compressed data are generally more prone to transmission errors than non-compressed data. The choice of a compression scheme must consider the error characteristics of the channel.
- Audio codecs need to be level calibrated so that when sound pressure level-dependent decisions are made in the coder those levels actually exist at the microphone.

Google Exhibit No. 1005
Page 038

Google Exhibit No. 1063
Page 0293

- Low bit rate coders should only be used for the final delivery of post-produced signals to the end-user.
- Don't believe statements comparing codec performance to 'VHS quality' or similar. Compression artifacts are quite different from the artifacts of consumer VCRs.
- Quality varies wildly with source material. Beware of 'convincing' demonstrations which may use selected material to achieve low bit rates. Use your own test material, selected for a balance of difficulty.
- Don't be browbeaten by the technology. You don't have to understand it to assess the results. Your eyes and ears are as good as anyone's so don't be afraid to criticize artifacts. In the case of video, use still frames to distinguish spatial artifacts from temporal ones.

References

1. ISO/IEC JTC1/SC29/WG11 MPEG, International standard ISO 11172, Coding of moving pictures and associated audio for digital storage media up to 1.5 Mbits/s (1992)
2. LeGall, D., MPEG: a video compression standard for multimedia applications. *Communications of the ACM*, 34, No.4, 46–58 (1991)
3. MPEG-2 Video Standard: ISO/IEC 13818–2: Information technology – generic coding of moving pictures and associated audio information: Video (1996) (aka ITU-T Rec. H-262 (1996))
4. MPEG-4 Standard: ISO/IEC 14496–2: Information technology – coding of audio-visual objects: Amd.1 (2000)
5. Huffman, D.A. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40 1098–1101 (1952)
6. ISO Joint Photographic Experts Group standard JPEG-8-R8
7. Wallace, G.K., Overview of the JPEG (ISO/CCITT) still image compression standard. ISO/JTC1/SC2/WG8 N932 (1989)

Google Exhibit No. 1005
Page 039

Google Exhibit No. 1063
Page 0294

performance of the coder. Generally, decoders do not cause significant loss of quality; they make the best of the data from the coder. Similarly, DACs cause little quality loss above that due to the ADC. In practical systems the loss of quality is dominated by the action of the coder. In communication theory, compression is known as *source coding* in order to distinguish it from the *channel coding* necessary reliably to send data down transmission or recording channels. This book is not concerned with channel coding but details can be found elsewhere.[2]

2.5 Sampling

Sampling is a process of periodic measurement which can take place in space or time and in several dimensions at once. Figure 2.4(a) shows that in *temporal sampling* the frequency of the signal to be sampled and the sampling rate F_s are measured in Hertz (Hz), the standard unit of temporal frequency. In still images such as photographs there is no temporal change and Figure 2.4(b) shows that the sampling is *spatial*. The sampling rate is now a spatial frequency. The absolute unit of spatial

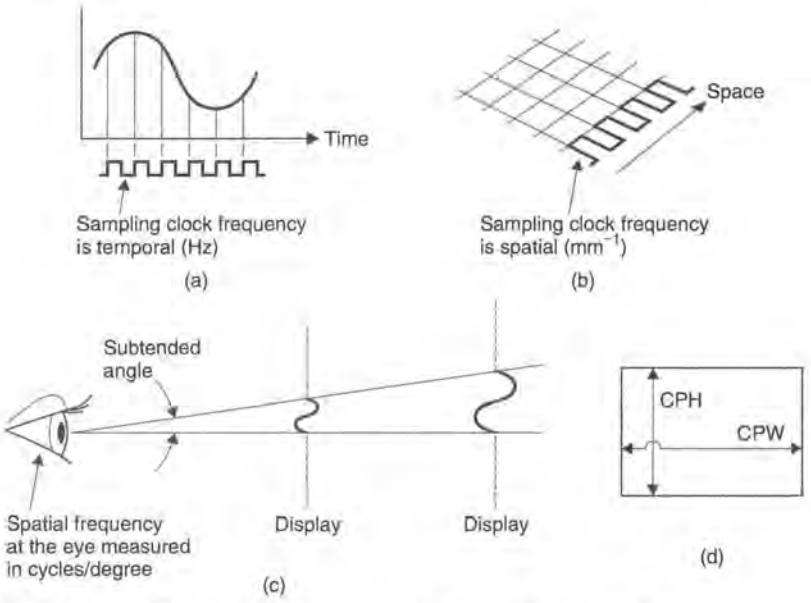


Figure 2.4 (a) Electrical waveforms are sampled temporally at a sampling rate measured in Hz. (b) Image information must be sampled spatially, but there is no single unit of spatial sampling frequency. (c) The acuity of the eye is measured as a subtended angle, and here two different displays of different resolutions give the same result at the eye because they are at a different distance. (d) Size-independent units such as cycles per picture height will also be found.

Google Exhibit No. 1005
Page 040

Google Exhibit No. 1063
Page 0295

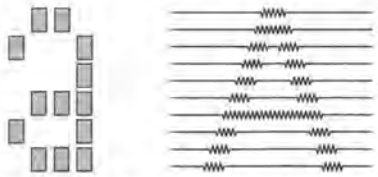


Figure 2.18 Some examples of sampled systems in which filtering is inadequate or absent.

little energy in the pulses and noise can be a problem. A practical limit is around 12.5 per cent where the frequency response is virtually ideal.

It should be stressed that in real systems there will often be more than one aperture effect. The result is that the frequency responses of the various aperture effects multiply, which is the same as saying that their impulse responses convolve. Whatever fine words are used, the result is an increasing loss of high frequencies where a series of acceptable devices when cascaded produce an unacceptable result.

In many systems, for reasons of economy or ignorance, reconstruction is simply not used and the system output is an unfiltered ZOH waveform. Figure 2.18 shows some examples of this kind of thing which are associated with the 'digital look'. It is important to appreciate that in well-engineered systems containing proper filters there is no such thing as the digital look.

2.8 Choice of audio sampling rate

The Nyquist criterion is only the beginning of the process which must be followed to arrive at a suitable sampling rate. The slope of available filters will compel designers to raise the sampling rate above the theoretical Nyquist rate. For consumer products, the lower the sampling rate, the better, since the cost of the medium or channel is directly proportional to the sampling rate: thus sampling rates near to twice 20 kHz are to be expected.

Where very low bit-rate compression is to be used, better results may be obtained by reducing the sampling rate so that the compression factor is not as great.

For professional products, there is a need to operate at variable speed for pitch correction. When the speed of a digital recorder is reduced, the offtape sampling rate falls, and Figure 2.19 shows that with a minimal sampling rate the first image frequency can become low enough to pass the reconstruction filter. If the sampling frequency is raised without changing the response of the filters, the speed can be reduced without this problem. It follows that variable-speed recorders, generally those with stationary heads, must use a higher sampling rate.

Google Exhibit No. 1005
Page 041

Google Exhibit No. 1063
Page 0296

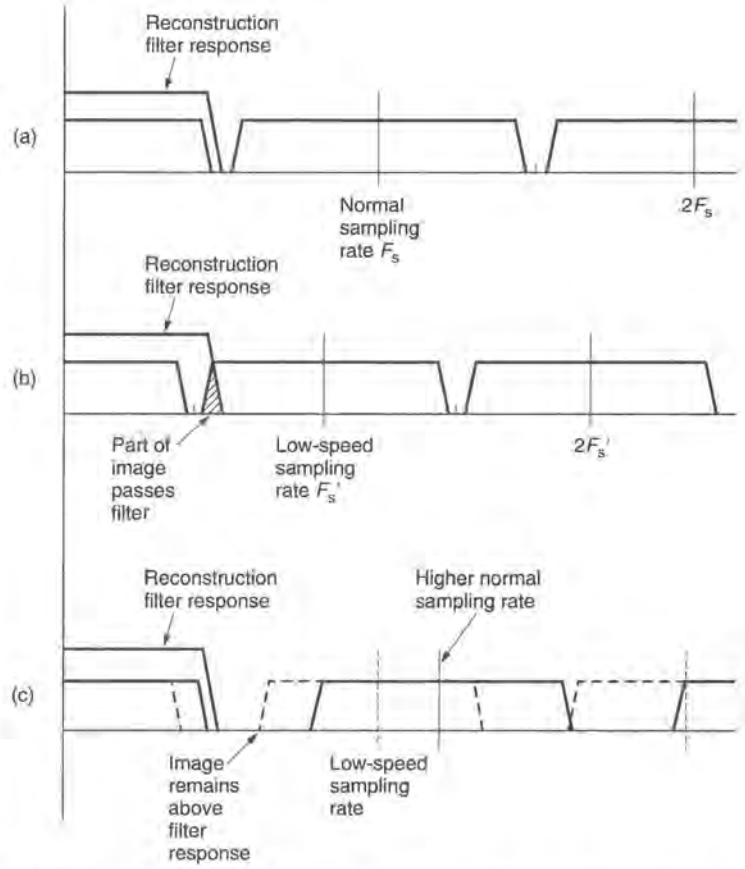


Figure 2.19 At normal speed, the reconstruction filter correctly prevents images entering the baseband, as in (a). When speed is reduced, the sampling rate falls, and a fixed filter will allow part of the lower sideband of the sampling frequency to pass. If the sampling rate of the machine is raised, but the filter characteristic remains the same the problem can be avoided, as in (c).

In the early days of digital audio, video recorders were adapted to store audio samples by creating a pseudo-video waveform which could convey binary as black and white levels.[9] The sampling rate of such a system is constrained to relate simply to the field rate and field structure of the television standard used, so that an integer number of samples can be stored on each usable TV line in the field. Such a recording can be made on a monochrome recorder, and these recordings are made in two standards, 525 lines at 60 Hz and 625 lines at 50 Hz. Thus it was necessary to find a frequency which is a common multiple of the two and also suitable for use as a sampling rate.

Google Exhibit No. 1005
Page 042

Google Exhibit No. 1063
Page 0297

The allowable sampling rates in a pseudo-video system can be deduced by multiplying the field rate by the number of active lines in a field (blanked lines cannot be used) and again by the number of samples in a line. By careful choice of parameters it is possible to use either 525/60 or 625/50 video with a sampling rate of 44.1 kHz.

In 60 Hz video, there are 35 blanked lines, leaving 490 lines per frame, or 245 lines per field for samples. If three samples are stored per line, the sampling rate becomes

$$60 \times 245 \times 3 = 44.1\,\text{kHz}$$

In 50 Hz video, there are 37 lines of blanking, leaving 588 active lines per frame, or 294 per field, so the same sampling rate is given by

$$50 \times 294 \times 3 = 44.1\,\text{kHz}$$

The sampling rate of 44.1 kHz came to be that of the Compact Disc. Even though CD has no video circuitry, the equipment used to make CD masters was originally video-based and determined the sampling rate.

For landlines to FM stereo broadcast transmitters having a 15 kHz audio bandwidth, the sampling rate of 32 kHz is more than adequate, and has been in use for some time in the United Kingdom and Japan. This frequency is also used in the NICAM 728 stereo TV sound system, in DVB audio and in DAB. The professional sampling rate of 48 kHz was proposed as having a simple relationship to 32 kHz, being far enough above 40 kHz for variable-speed operation, and having a simple relationship with 50 Hz frame rate video which would allow digital video recorders to store the convenient number of 960 audio samples per video field. This is the sampling rate used by all production DVTRs. The field rate offset of 59.94 Hz video does not easily relate to any of the above sampling rates, and requires special handling which is outside the scope of this book.[10]

Although in a perfect world the adoption of a single sampling rate might have had virtues, for practical and economic reasons digital audio now has essentially three rates to support: 32 kHz for broadcast, 44.1 kHz for CD, and 48 kHz for professional use.[11] In MPEG these audio sampling rates may be halved for low bit rate applications with a corresponding loss of audio bandwidth.

2.9 Video sampling structures

Component or colour difference signals are used primarily for post-production work where quality and flexibility are paramount. In colour

Google Exhibit No. 1005
Page 043

Google Exhibit No. 1063
Page 0298

difference working, the important requirement is for image manipulation in the digital domain. This is facilitated by a sampling rate which is a multiple of line rate because then there is a whole number of samples in a line and samples are always in the same position along the line and can form neat columns. A practical difficulty is that the line period of the 525 and 625 systems is slightly different. The problem was overcome by the use of a sampling clock which is an integer multiple of both line rates.

ITU-601 (formerly CCIR-601) recommends the use of certain sampling rates which are based on integer multiples of the carefully chosen fundamental frequency of 3.375 MHz. This frequency is normalized to 1 in the document.

In order to sample 625/50 luminance signals without quality loss, the lowest multiple possible is 4 which represents a sampling rate of 13.5 MHz. This frequency line-locks to give 858 samples per line period in 525/59.94 and 864 samples per line period in 625/50.

In the component analog domain, the colour difference signals used for production purposes typically have one half the bandwidth of the luminance signal. Thus a sampling rate multiple of 2 is used and results in 6.75 MHz. This sampling rate allows respectively 429 and 432 samples per line. Component video sampled in this way has a 4:2:2 format. Whilst other combinations are possible, 4:2:2 is the format for which the majority of digital component production equipment is constructed. The D-1, D-5, D-9, SX and Digital Betacam DVTRs operate with 4:2:2 format data. Figure 2.20 shows the spatial arrangement given by 4:2:2 sampling.

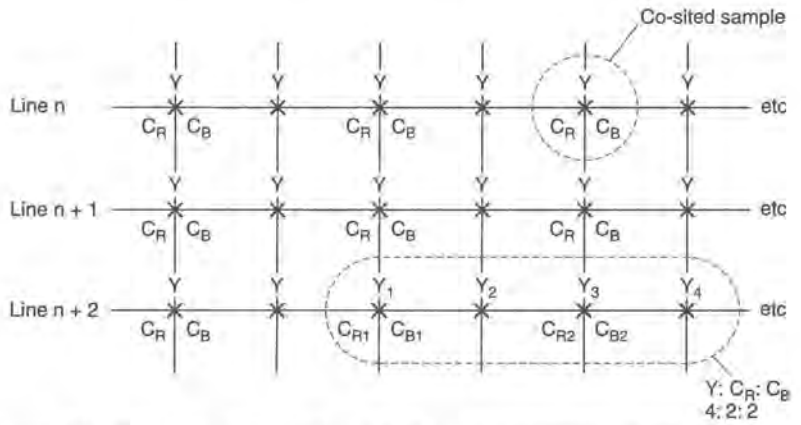


Figure 2.20 In CCIR-601 sampling mode 4:2:2, the line synchronous sampling rate of 13.5 MHz results in samples having the same position in successive lines, so that vertical columns are generated. The sampling rates of the colour difference signals C_R, C_B are one-half of that of luminance, i.e. 6.75 MHz, so that there are alternate Y only samples and co-sited samples which describe Y, C_R and C_B. In a run of four samples, there will be four Y samples, two C_R samples and two C_B samples, hence 4:2:2.

Google Exhibit No. 1005
Page 044

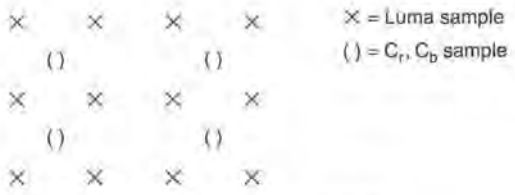
Google Exhibit No. 1063
Page 0299

Luminance samples appear at half the spacing of colour difference samples, and every other luminance sample is co-sited with a pair of colour difference samples. Co-siting is important because it allows all attributes of one picture point to be conveyed with a three-sample vector quantity. Modification of the three samples allows such techniques as colour correction to be performed. This would be difficult without co-sited information. Co-siting is achieved by clocking the three ADCs simultaneously.

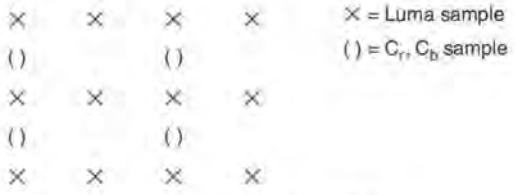
For lower bandwidths, particularly in prefiltering operations prior to compression, the sampling rate of the colour difference signal can be halved. 4:1:1 delivers colour bandwidth in excess of that required by the composite formats and is used in 60 Hz DV camcorder formats.

In 4:2:2 the colour difference signals are sampled horizontally at half the luminance sampling rate, yet the vertical colour difference sampling rates are the same as for luma. Whilst this is not a problem in a production application, this disparity of sampling rates represents a data rate overhead which is undesirable in a compression environment. In this case it is possible to halve the vertical sampling rate of the colour difference signals as well, producing a format known as 4:2:0.

This topic is the source of considerable confusion. In MPEG-1, which was designed for a low bit rate, the single ideal 4:2:0 subsampling strategy of (Figure 2.21(a) was used. The colour data are vertically low-pass filtered to the same bandwidth in two dimensions and interpolated



(a) MPEG-1 chroma downsampling (4:2:0)



(b) MPEG-2 chroma downsampling (4:2:0)

Figure 2.21 (a) In MPEG-1, colour difference information is downsampled and shifted so that it is symmetrically disposed about the luminance pixels. (b) In MPEG-2 the horizontal shift was abandoned.

Google Exhibit No. 1005
Page 045

Google Exhibit No. 1063
Page 0300

so that the remaining colour pixels are equidistant from the source pixels. This has the effect of symmetrically disposing the colour information with respect to the luma. When MPEG-2 was developed, it was a requirement to support 4:4:4, 4:2:2 and 4:2:0 colour structures. Figure 2.21(b) shows that in MPEG-2, the colour difference samples require no horizontal interpolation in 4:4:4 or 4:2:2 and so for consistency they don't get it in 4:2:0 either. As a result there are two different 4:2:0 structures, one for MPEG-1 and one for MPEG-2 and MPEG-4. In vertical subsampling a new virtual raster has been created for the chroma samples. At the decoder a further interpolation will be required to put the decoded chroma data back onto the display raster. This causes a small generation loss which is acceptable for a single-generation codec as used in broadcasting, but not in multi-generation codecs needed for production. This problem was one of the reasons for the development of the 4:2:2 Profile of MPEG-2 which avoids the problem by retaining the colour information on every line.

The sampling rates of ITU-601 are based on commonality between 525- and 625-line systems. However, the consequence is that the pixel spacing is different in the horizontal and vertical axes. This is incompatible with computer graphics in which so-called 'square' pixels are used. This means that the horizontal and vertical spacing is the same, giving the same resolution in both axes. However, high-definition TV and computer graphics formats universally use 'square' pixels. MPEG can handle various pixel aspect ratios and allows a control code to be embedded in the sequence header to help the decoder.

2.10 The phase-locked loop

All digital video systems need to be clocked at the appropriate rate in order to function properly. Whilst a clock may be obtained from a fixed frequency oscillator such as a crystal, many operations in video require *genlocking* or synchronizing the clock to an external source.

In phase-locked loops, the oscillator can run at a range of frequencies according to the voltage applied to a control terminal. This is called a voltage-controlled oscillator or VCO. Figure 2.22 shows that the VCO is driven by a phase error measured between the output and some reference. The error changes the control voltage in such a way that the error is reduced, so that the output eventually has the same frequency as the reference. A low-pass filter is fitted in the control voltage path to prevent the loop becoming unstable. If a divider is placed between the VCO and the phase comparator, as in the figure, the VCO frequency can be made to be a multiple of the reference. This also has the effect of making the loop more heavily damped, so that it is less likely to change frequency if the input is irregular.

Google Exhibit No. 1005
Page 046

Google Exhibit No. 1063
Page 0301

5

MPEG video compression

In this chapter the principles of video compression are explored, leading to descriptions of MPEG-1, MPEG-2 and MPEG-4. MPEG-1 supports only progressively scanned images, whereas MPEG-2 and MPEG-4 support both progressive and interlaced scan. MPEG uses the term 'picture' to mean a full-screen image of any kind at one point on the time axis. This could be a field or a frame in interlaced systems but only a frame in non-interlaced systems. The terms field and frame will be used only when the distinction is important. MPEG-4 introduces object coding which can handle entities that may not fill the screen. In MPEG-4 the picture becomes a plane in which one or more video objects can be displayed, hence the term video object plane (VOP).

5.1 The eye

All imaging signals ultimately excite some response in the eye and the viewer can only describe the result subjectively. Familiarity with the functioning and limitations of the eye is essential to an understanding of image compression. The simple representation of Figure 5.1 shows that the eyeball is nearly spherical and is swivelled by muscles. The space between the cornea and the lens is filled with transparent fluid known as *aqueous humour*. The remainder of the eyeball is filled with a transparent jelly known as *vitreous humour*. Light enters the cornea, and the amount of light admitted is controlled by the pupil in the iris. Light entering is involuntarily focused on the retina by the lens in a process called *visual accommodation*. The lens is the only part of the eye which is not nourished by the bloodstream and its centre is technically dead. In a young person the lens is flexible and muscles distort it to perform the focusing action

Google Exhibit No. 1005
Page 047

Google Exhibit No. 1063
Page 0302

developers at a time when things were so much harder. However, we must also recognize that the context in which interlace made sense has disappeared.

5.7 Spatial and temporal redundancy in MPEG

Chapter 1 introduced these concepts in a general sense and now they will be treated with specific reference to MPEG. Figure 5.18(a) shows that spatial redundancy is redundancy within a single picture or object, for example repeated pixel values in a large area of blue sky. Temporal redundancy (b) exists between successive pictures or objects.

In MPEG, where temporal compression is used, the current picture/object is not sent in its entirety; instead the difference between the current picture/object and the previous one is sent. The decoder already has the previous picture/object, and so it can add the difference, or residual image, to make the current picture/object. A residual image is created by subtracting every pixel in one picture/object from the corresponding pixel in another. This is trivially easy when pictures are restricted to progressive scan, as in MPEG-1, but MPEG-2 had to develop greater complexity (continued in MPEG-4) so that this can also be done with interlaced pictures. The handling of interlace in MPEG will be detailed later.

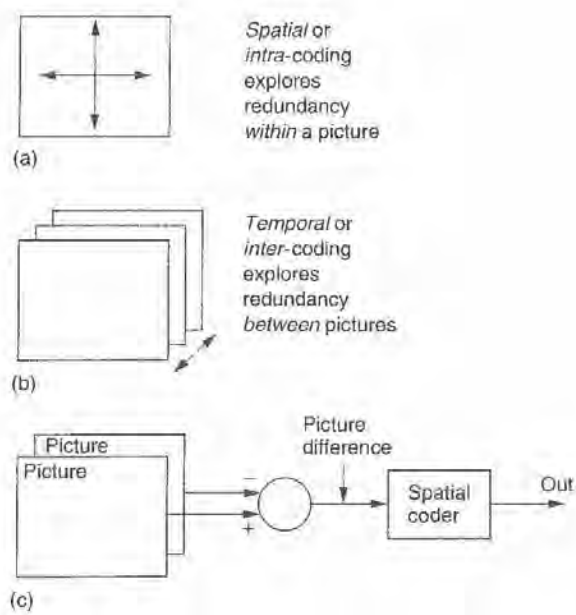


Figure 5.18 (a) Spatial or intra-coding works on individual images. (b) Temporal or inter-coding works on successive images. (c) In MPEG inter-coding is used to create difference images. These are then compressed spatially.

Google Exhibit No. 1005
Page 048

Google Exhibit No. 1063
Page 0303

A residual is an image of a kind, although not a viewable one, and so should contain some kind of spatial redundancy. Figure 5.18(c) shows that MPEG takes advantage of both forms of redundancy. Residual images are spatially compressed prior to transmission. At the decoder the spatial compression is decoded to re-create the residual, then this is added to the previous picture/object to complete the decoding process.

Whenever objects move they will be in a different place in successive pictures. This will result in large amounts of difference data. MPEG overcomes the problem using motion compensation. The encoder contains a motion estimator which measures the direction and distance of motion between pictures and outputs these as vectors which are sent to the decoder. When the decoder receives the vectors it uses them to shift data in a previous picture to more closely resemble the current picture. Effectively the vectors are describing the optic flow axis of some moving screen area, along which axis the image is highly redundant. Vectors are bipolar codes which determine the amount of horizontal and vertical shift required.

In real images, moving objects do not necessarily maintain their appearance as they move. For example, objects may move into shade or light. Consequently motion compensation can never be ideal and it is still necessary to send a residual to make up for any shortcomings in the motion compensation.

Figure 5.19 shows how this works in MPEG-1 and MPEG-2. In addition to the motion-encoding system, the coder also contains a motion decoder. When the encoder outputs motion vectors, it also uses them locally in the

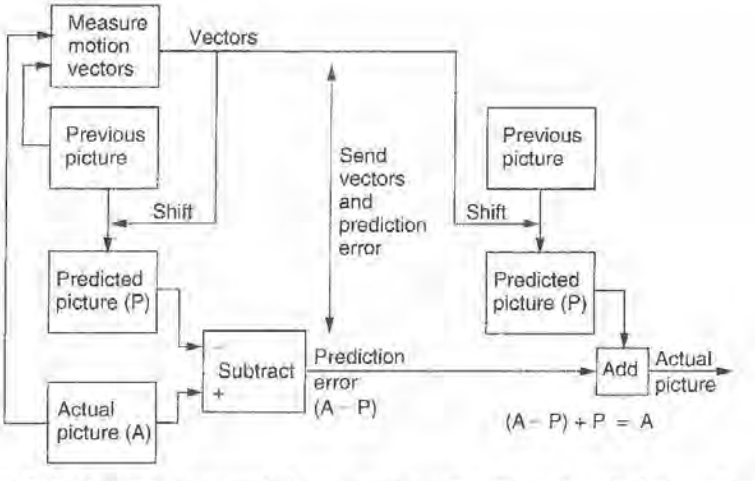


Figure 5.19 A motion-compensated compression system. The coder calculates motion vectors which are transmitted as well as being used locally to create a predicted picture. The difference between the predicted picture and the actual picture is transmitted as a prediction error.

Google Exhibit No. 1005
Page 049

Google Exhibit No. 1063
Page 0304

same way that a real decoder will, and is able to produce a *predicted picture* based solely on the previous picture shifted by motion vectors. This is then subtracted from the *actual* current picture to produce a *prediction error* or *residual* which is an image of a kind that can be spatially compressed. The decoder takes the previous picture, shifts it with the vectors to re-create the predicted picture and then decodes and adds the prediction error to produce the actual picture. Picture data sent as vectors plus prediction error are said to be *P* coded.

The concept of sending a prediction error is a useful approach because it allows both the motion estimation and compensation to be imperfect. An ideal motion compensation system will send just the right amount of vector data. With insufficient vector data, the prediction error will be large, but transmission of excess vector data will also cause the the bit rate to rise. There will be an optimum balance which minimizes the sum of the prediction error data and the vector data.

In MPEG-1 and MPEG-2 the balance is obtained by dividing the screen into areas called *macroblocks* which are 16 luminance pixels square. Each macroblock is associated with a vector. The vector has horizontal and vertical components so that, used in combination, these can shift pixel data in any direction. The location of the boundaries of a macroblock are fixed with respect to the display screen and so clearly the vector cannot move the macroblock. Instead the vector tells the decoder where to look in another picture to find pixel data which will be fetched *to* the macroblock. Figure 5.20(a) shows this concept. The shifting process is generally done by modifying the read address of a RAM using the vector. This can shift by one-pixel steps. MPEG-2 vectors have half-pixel resolution so it is necessary to interpolate between pixels from RAM to obtain half-pixel shifted values. Chapter 3 showed how interpolations of this kind can be performed.

MPEG-1 and MPEG-2 also use a simple compression scheme on the vectors. Figure 5.21 shows that in the case of a large moving object, all the vectors within the object will be identical or nearly so. A horizontal run of macroblocks can be associated to form a structure called a *slice* (see section 5.14). The first vector of a slice is sent as an absolute value, but subsequent vectors will be sent as differences. No advantage is taken of vertical vector redundancy.

Real moving objects will not coincide with macroblocks and so the motion compensation will not be ideal but the prediction error makes up for any shortcomings. Figure 5.20(b) shows the case where the boundary of a moving object bisects a macroblock. If the system measures the moving part of the macroblock and sends a vector, the decoder will shift the entire block making the stationary part wrong. If no vector is sent, the moving part will be wrong. Both approaches are legal in MPEG-1 and MPEG-2 because the prediction error compensates for the incorrect

Google Exhibit No. 1005
Page 050

Google Exhibit No. 1063
Page 0305

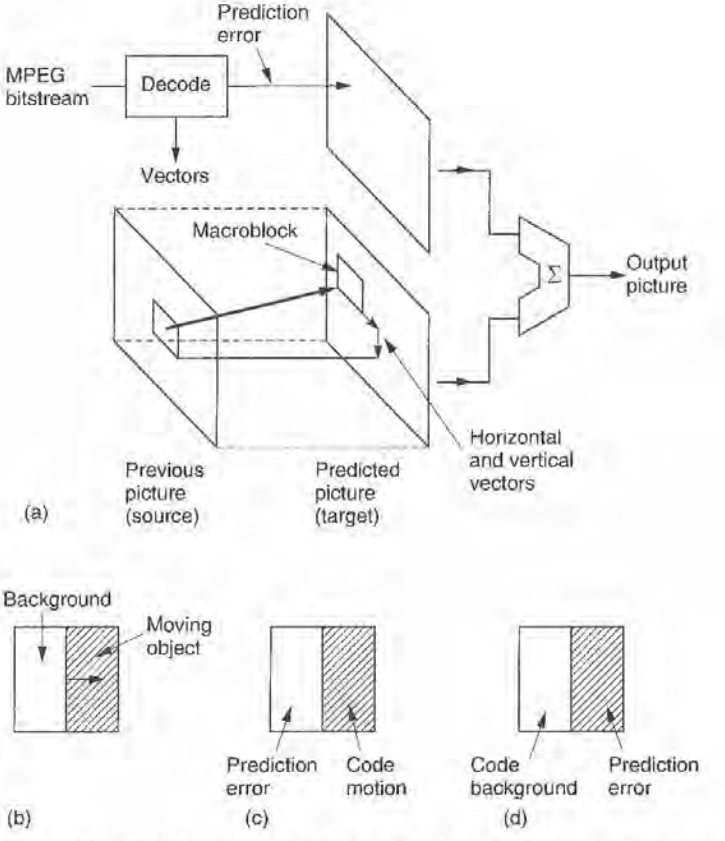


Figure 5.20 (a) In motion compensation, pixel data are brought to a fixed macroblock in the target picture from a variety of places in another picture. (b) Where only part of a macroblock is moving, motion compensation is non-ideal. The motion can be coded (c), causing a prediction error in the background, or the background can be coded (d) causing a prediction error in the moving object.

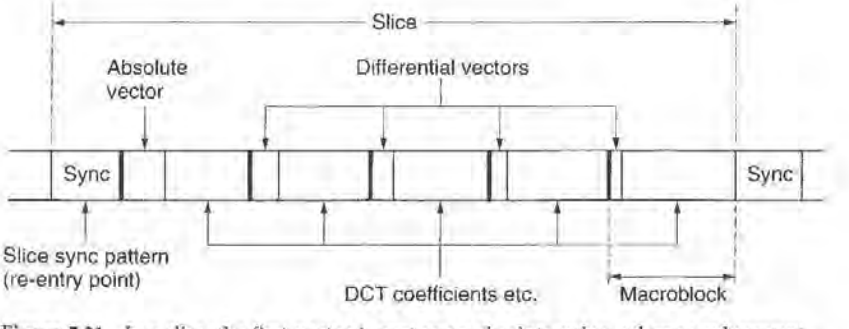


Figure 5.21 In a slice, the first vector is sent as an absolute value, whereas subsequent vectors are sent as differences. When a large moving object is to be coded, several macroblocks may have the same velocity and this is efficiently coded in a slice.

Google Exhibit No. 1005
Page 051

Google Exhibit No. 1063
Page 0306

values. An intelligent coder might try both approaches to see which required the least prediction error data.

The prediction error concept also allows the use of simple but inaccurate motion estimators in low-cost systems. The greater prediction error data are handled using a higher bit rate. On the other hand if a precision motion estimator is available, a very high compression factor may be achieved because the prediction error data are minimized. MPEG does not specify how motion is to be measured; it simply defines how a decoder will interpret the vectors. Encoder designers are free to use any motion-estimation system provided that the right vector protocol is created. Chapter 3 contrasted a number of motion estimation techniques.

Figure 5.22(a) shows that a macroblock contains both luminance and colour difference data at different resolutions. Most of the MPEG-2 Profiles use a 4:2:0 structure which means that the colour is downsampled by a factor of two in both axes. Thus in a 16 × 16 pixel block, there are only 8 × 8 colour difference sampling sites. MPEG-2 is based upon the 8 × 8 DCT (see section 3.7) and so the 16 × 16 block is the screen area which contains an 8 × 8 colour difference sampling block. Thus in 4:2:0 in each macroblock there are four luminance DCT blocks, one $R-Y$ DCT block and one $B-Y$ DCT block, all steered by the same vector.

In the 4:2:2 Profile of MPEG-2, shown in Figure 5.22(b), the chroma is not downsampled vertically, and so there is twice as much chroma data in each macroblock which is otherwise substantially the same.

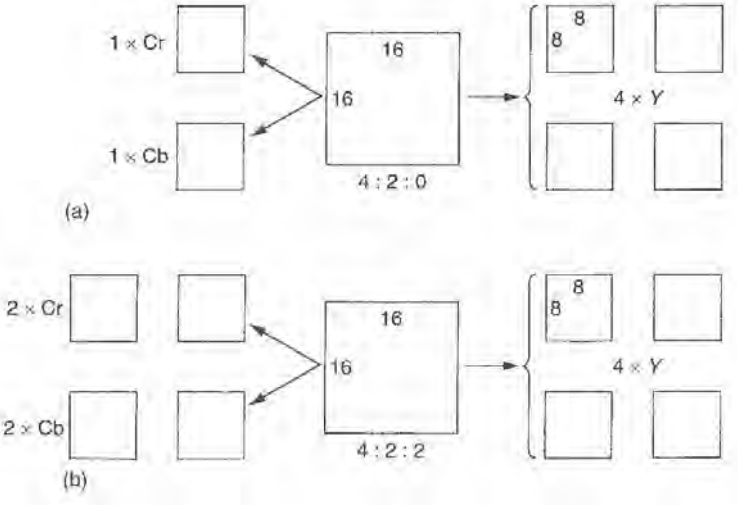


Figure 5.22 The structure of a macroblock. (A macroblock is the screen area steered by *one* vector.) (a) In 4:2:0, there are two chroma DCT blocks per macroblock whereas in 4:2:2 (b) there are four, 4:2:2 needs 33 per cent more data than 4:2:0.

Google Exhibit No. 1005
Page 052

Google Exhibit No. 1063
Page 0307

In MPEG-4 a number of refinements are made to motion compensation in addition to the MPEG-2 tools which are still available. One of these tools is a more complex form of vector compression which will be described in section 5.21. This cuts down the bit rate needed for a given set of vector data and has a number of benefits. By retaining the existing macroblock size, a higher compression factor results from the reduced vector data. However, the opportunity then arises to increase the vector density. Optionally, MPEG-4 allows one vector to be associated with an 8 × 8 DCT block rather than with a 16 × 16 macroblock. This allows much more precise prediction of complex motion, especially near the edge of objects, and will result in a reduced residual bit rate. On a macroblock-by-macroblock basis an MPEG-4 coder can decide whether to code four vectors per macroblock or only one. When the compressed vector data and residual are combined, one of these approaches will result in the least data.

5.8 *I* and *P* coding

Predictive (*P*) coding cannot be used indefinitely, as it is prone to error propagation. A further problem is that it becomes impossible to decode the transmission if reception begins part-way through. In real video signals, cuts or edits can be present across which there is little redundancy and which make motion estimators throw up their hands.

In the absence of redundancy over a cut, there is nothing to be done but to send the new picture information in absolute form. This is called *I* coding where *I* is an abbreviation of *intra* coding. As *I* coding needs no previous picture for decoding, then decoding can begin at *I* coded information.

MPEG is effectively a toolkit. The bigger the MPEG-number, the more tools are available. However, there is no compulsion to use all the tools available. Thus an encoder may choose whether to use *I* or *P* coding, either once and for all or dynamically on a macroblock-by-macroblock basis. For practical reasons, an entire frame may be encoded as *I* macroblocks periodically. This creates a place where the bitstream might be edited or where decoding could begin.

Figure 5.23 shows a typical application of the Simple Profile of MPEG-2. Periodically an *I* picture is created. Between *I* pictures are *P* pictures which are based on the picture before. These *P* pictures predominantly contain macroblocks having vectors and prediction errors. However, it is perfectly legal for *P* pictures to contain *I* macroblocks. This might be useful where, for example, a camera pan introduces new material at the edge of the screen which cannot be created from an earlier picture.

Google Exhibit No. 1005
Page 053

Google Exhibit No. 1063
Page 0308

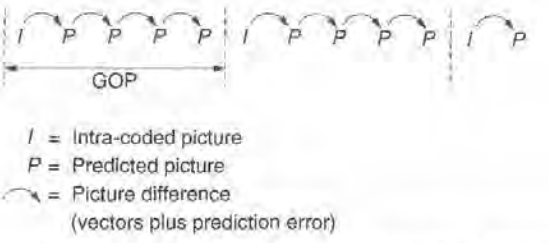


Figure 5.23 A Simple Profile MPEG-2 signal may contain periodic *I* pictures with a number of *P* pictures between.

Note that although what is sent is called a *P* picture, it is not a picture at all. It is a set of instructions to convert the previous picture into the current picture. If the previous picture is lost, decoding is impossible.

An *I* picture together with all of the pictures before the next *I* picture form a *group of pictures* (GOP).

5.9 Bidirectional coding

Motion-compensated predictive coding is a useful compression technique, but it does have the drawback that it can only take data from a previous picture.

Where moving objects reveal a background this is completely unknown in previous pictures and forward prediction fails. However, more of the background is visible in later pictures. Figure 5.24 shows the concept. In

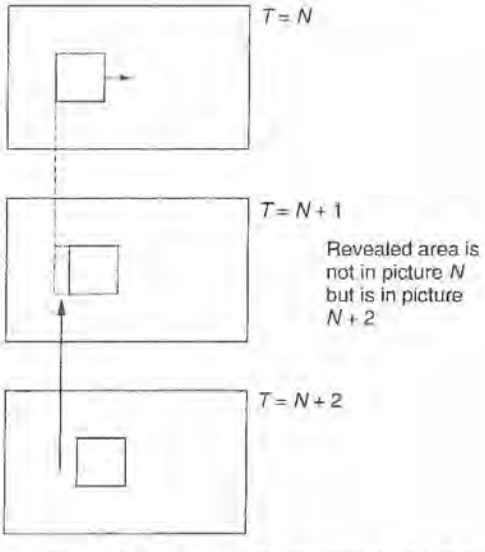


Figure 5.24 In bidirectional coding the revealed background can be efficiently coded by bringing data back from a future picture.

Google Exhibit No. 1005
Page 054

Google Exhibit No. 1063
Page 0309

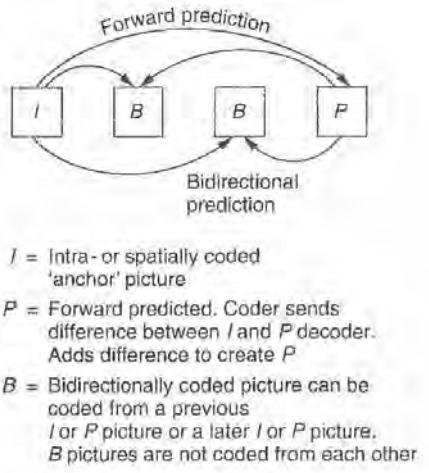


Figure 5.25 In bidirectional coding, a number of *B* pictures can be inserted between periodic forward predicted pictures. See text.

the centre of the diagram, a moving object has revealed some background. The previous picture can contribute nothing, whereas the next picture contains all that is required.

Bidirectional coding is shown in Figure 5.25. A bidirectional or *B* macroblock can be created using a combination of motion compensation and the addition of a prediction error. This can be done by forward prediction from a previous picture or backward prediction from a subsequent picture. It is also possible to use an average of both forward and backward prediction. On noisy material this may result in some reduction in bit rate. The technique is also a useful way of portraying a dissolve.

The averaging process in MPEG is a simple linear interpolation which works well when only one *B* picture exists between the reference pictures before and after. A larger number of *B* pictures would require weighted interpolation but MPEG does not support this.

Typically two *B* pictures are inserted between *P* pictures or between *I* and *P* pictures. As can be seen, *B* pictures are never predicted from one another, only from *I* or *P* pictures. A typical GOP for broadcasting purposes might have the structure *IBBPBBPBBPBB*. Note that the last *B* pictures in the GOP require the *I* picture in the next GOP for decoding and so the GOPs are not truly independent. Independence can be obtained by creating a *closed GOP* which may contain *B* pictures but which ends with a *P* picture. It is also legal to have a *B* picture in which every macroblock is forward predicted, needing no future picture for decoding.

Bidirectional coding is very powerful. Figure 5.26 is a constant quality curve showing how the bit rate changes with the type of coding. On the

Google Exhibit No. 1005
Page 055

Google Exhibit No. 1063
Page 0310

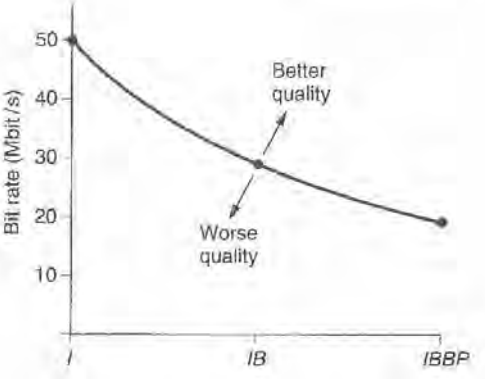


Figure 5.26 Bidirectional coding is very powerful as it allows the same quality with only 40 per cent of the bit rate of intra-coding. However, the encoding and decoding delays must increase. Coding over a longer time span is more efficient but editing is more difficult.

left, only *I* or spatial coding is used, whereas on the right a *IBBP* structure is used. This means that there are two bidirectionally coded pictures in between a spatially coded picture (*I*) and a forward predicted picture (*P*). Note how for the same quality the system which only uses spatial coding needs two and a half times the bit rate that the bidirectionally coded system needs.

Clearly information in the future has yet to be transmitted and so is not normally available to the decoder. MPEG gets around the problem by sending pictures in the wrong order. Picture reordering requires delay in the encoder and a delay in the decoder to put the order right again. Thus the overall codec delay must rise when bidirectional coding is used. This is quite consistent with Figure 1.5 which showed that as the compression factor rises the latency must also rise.

Figure 5.27 shows that although the original picture sequence is *IBBPBBPBBIBB* ..., this is transmitted as *IPBBPBBIBB* ... so that the future picture is already in the decoder before bidirectional decoding begins. Note that the *I* picture of the next GOP is actually sent before the last *B* pictures of the current GOP.

Figure 5.27 also shows that the amount of data required by each picture is dramatically different. *I* pictures have only spatial redundancy and so need a lot of data to describe them. *P* pictures need fewer data because they are created by shifting the *I* picture with vectors and then adding a prediction error picture. *B* pictures need the least data of all because they can be created from *I* or *P*.

With pictures requiring a variable length of time to transmit, arriving in the wrong order, the decoder needs some help. This takes the form of picture-type flags and time stamps which will be described in section 6.2.

Google Exhibit No. 1005
Page 056

Google Exhibit No. 1063
Page 0311

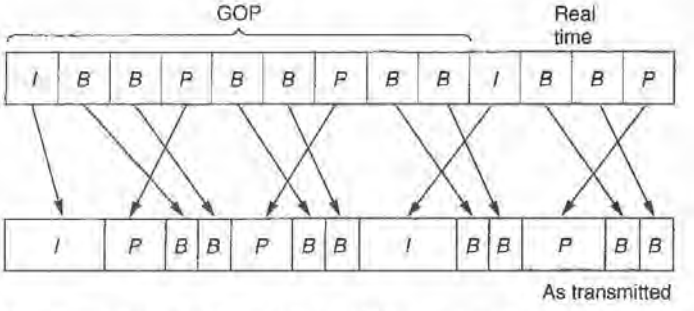


Figure 5.27 Comparison of pictures before and after compression showing sequence change and varying amount of data needed by each picture type. *I, P, B* pictures use unequal amounts of data.

5.10 Coding applications

Figure 5.28 shows a variety of possible GOP (group of pictures in MPEG-1 and MPEG-2) or GOV (group of video object planes in MPEG-4) structures. The simplest is the *III* . . . sequence in which every picture (or object in MPEG-4) is intra-coded. These can be fully decoded without reference to any other picture or object and so editing is straightforward. However, this approach requires about two-and-one-half times the bit rate of a full bidirectional system.

Bidirectional coding is most useful for final delivery of post-produced material either by broadcast or on prerecorded media as there is then no editing requirement. As a compromise the *IBIB* . . . structure can be used which has some of the bit rate advantage of bidirectional coding but without too much latency. It is possible to edit an *IBIB* stream by performing some processing. If it is required to remove the video following a *B* picture, that *B* picture could not be decoded because it needs *I* pictures either side of it for bidirectional decoding. The solution

I I I I I I ...	*I* only freely editable, needs high bit rate
I P P P P I P ...	Forward predicted only, needs less decoder memory, used in Simple Profile
I B B P B B P B ...	Forward and bidirectional, best compression factor, needs large decoder memory, hard to edit
I B I B I B ...	Lower bit rate than *I* only, editable with moderate processing

Figure 5.28 Various possible GOP structures used with MPEG. See text for details.

Google Exhibit No. 1005
Page 057

Google Exhibit No. 1063
Page 0312

is to decode the *B* picture first, and then re-encode it with forward prediction only from the previous *I* picture. The subsequent *I* picture can then be replaced by an edit process. Some quality loss is inevitable in this process but this is acceptable in applications such as ENG and industrial video.

5.11 Intra-coding

Intra-coding or spatial compression in MPEG is used in *I* pictures on actual picture data and in *P* and *B* pictures on prediction error data. MPEG-1 and MPEG-2 use the discrete cosine transform described in section 3.13. In still pictures, MPEG-4 may also use the wavelet transform described in section 3.14.

Entering the spatial frequency domain has two main advantages. It allows dominant spatial frequencies which occur in real images to be efficiently coded, and it allows noise shaping to be used. As the eye is not uniformly sensitive to noise at all spatial frequencies, dividing the information up into frequency bands allows a different noise level to be produced in each.

The DCT works on blocks and in MPEG these are 8×8 pixels. Section 5.7 showed how the macroblocks of the motion compensation structure are designed so they can be broken down into 8×8 DCT blocks. In a 4:2:0 macroblock there will be six DCT blocks whereas in a 4:2:2 macroblock there will be eight.

Figure 5.29 shows the table of basis functions or *wave table* for an 8×8 DCT. Adding these two-dimensional waveforms together in different proportions will give any original 8×8 pixel block. The coefficients of the DCT simply control the proportion of each wave which is added in the inverse transform. The top-left wave has no modulation at all because it conveys the DC component of the block. This coefficient will be a unipolar (positive only) value in the case of luminance and will typically be the largest value in the block as the spectrum of typical video signals is dominated by the DC component.

Increasing the DC coefficient adds a constant amount to every pixel. Moving to the right the coefficients represent increasing horizontal spatial frequencies and moving downwards the coefficients represent increasing vertical spatial frequencies. The bottom-right coefficient represents the highest diagonal frequencies in the block. All these coefficients are bipolar, where the polarity indicates whether the original spatial waveform at that frequency was inverted.

Figure 5.30 shows a one-dimensional example of an inverse transform. The DC coefficient produces a constant level throughout the pixel block. The remaining waves in the table are AC coefficients. A zero coefficient

Google Exhibit No. 1005
Page 058

Google Exhibit No. 1063
Page 0313

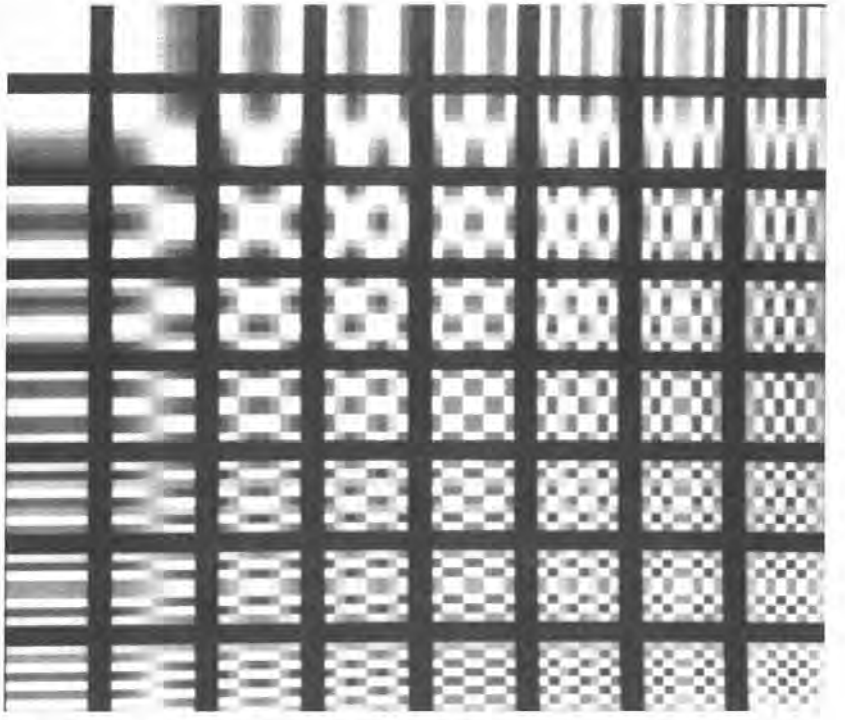


Figure 5.29 The discrete cosine transform breaks up an image area into discrete frequencies in two dimensions. The lowest frequency can be seen here at the top-left corner. Horizontal frequency increases to the right and vertical frequency increases downwards.

would result in no modulation, leaving the DC level unchanged. The wave next to the DC component represents the lowest frequency in the transform which is half a cycle per block. A positive coefficient would make the left side of the block brighter and the right side darker whereas a negative coefficient would do the opposite. The magnitude of the coefficient determines the amplitude of the wave which is added. Figure 5.30 also shows that the next wave has a frequency of one cycle per block. i.e. the block is made brighter at both sides and darker in the middle. Consequently an inverse DCT is no more than a process of mixing various pixel patterns from the wave table where the relative amplitudes and polarity of these patterns are controlled by the coefficients. The original transform is simply a mechanism which finds the coefficient amplitudes from the original pixel block.

The DCT itself achieves no compression at all. Sixty-four pixels are converted to sixty-four coefficients. However, in typical pictures, not all coefficients will have significant values; there will often be a few dominant coefficients. The coefficients representing the higher two-

Google Exhibit No. 1005
Page 059

Google Exhibit No. 1063
Page 0314

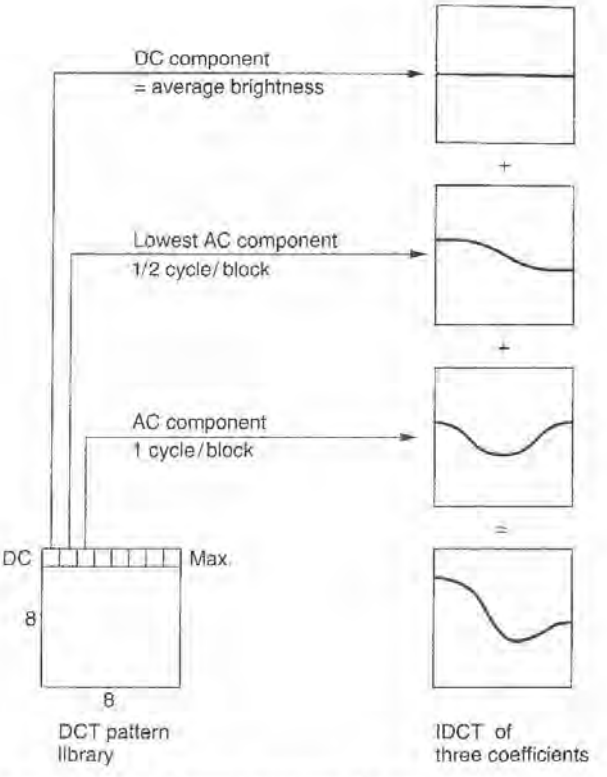


Figure 5.30 A one-dimensional inverse transform. See text for details.

dimensional spatial frequencies will often be zero or of small value in large areas, due to blurring or simply plain undetailed areas before the camera.

Statistically, the further from the top-left corner of the wave table the coefficient is, the smaller will be its magnitude. Coding gain (the technical term for reduction in the number of bits needed) is achieved by transmitting the low-valued coefficients with shorter wordlengths. The zero-valued coefficients need not be transmitted at all. Thus it is not the DCT which compresses the data, it is the subsequent processing. The DCT simply expresses the data in a form which makes the subsequent processing easier.

Higher compression factors require the coefficient wordlength to be further reduced using requantizing. Coefficients are divided by some factor which increases the size of the quantizing step. The smaller number of steps which results permits coding with fewer bits, but of course with an increased quantizing error. The coefficients will be multiplied by a reciprocal factor in the decoder to return to the correct magnitude.

Google Exhibit No. 1005
Page 060

Google Exhibit No. 1063
Page 0315

Inverse transforming a requantized coefficient means that the frequency it represents is reproduced in the output with the wrong amplitude. The difference between original and reconstructed amplitude is regarded as a noise added to the wanted data. Figure 5.10 showed that the visibility of such noise is far from uniform. The maximum sensitivity is found at DC and falls thereafter. As a result the top-left coefficient is often treated as a special case and left unchanged. It may warrant more error protection than other coefficients.

Transform coding takes advantage of the falling sensitivity to noise. Prior to requantizing, each coefficient is divided by a different weighting constant as a function of its frequency. Figure 5.31 shows a typical weighting process. Naturally the decoder must have a corresponding inverse weighting. This weighting process has the effect of reducing the magnitude of high-frequency coefficients disproportionately. Clearly different weighting will be needed for colour difference data as colour is perceived differently.

P and *B* pictures are decoded by adding a prediction error image to a reference image. That reference image will contain weighted noise. One purpose of the prediction error is to cancel that noise to prevent tolerance build-up. If the prediction error were also to contain weighted noise this result would not be obtained. Consequently prediction error coefficients are flat weighted.

When forward prediction fails, such as in the case of new material introduced in a *P* picture by a pan, *P* coding would set the vectors to zero

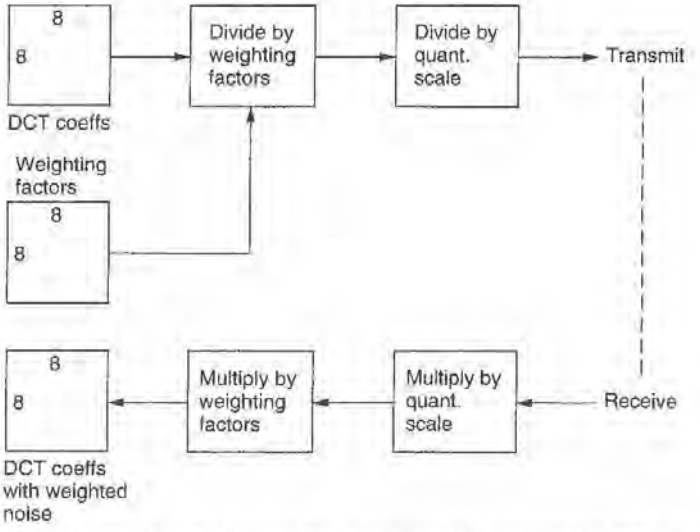


Figure 5.31 Weighting is used to make the noise caused by requantizing different at each frequency.

Google Exhibit No. 1005
Page 061

Google Exhibit No. 1063
Page 0316

and encode the new data entirely as an unweighted prediction error. In this case it is better to encode that material as an *I* macroblock because then weighting can be used and this will require fewer bits.

Requantizing increases the step size of the coefficients, but the inverse weighting in the decoder results in step sizes which increase with frequency. The larger step size increases the quantizing noise at high frequencies where it is less visible. Effectively the noise floor is shaped to match the sensitivity of the eye. The quantizing table in use at the encoder can be transmitted to the decoder periodically in the bitstream.

5.12 Intra-coding in MPEG-1 and MPEG-2

Study of the signal statistics gained from extensive analysis of real material is used to measure the probability of a given coefficient having a given value. This probability turns out to be highly non-uniform suggesting the possibility of a variable-length encoding for the coefficient values. On average, the higher the spatial frequency, the lower the value of a coefficient will be. This means that the value of a coefficient tends to fall as a function of its radius from the DC coefficient. DC coefficients also have certain characteristics. As they represent the average brightness of a block, in many picture areas, adjacent blocks will have similar DC coefficient values. This can be exploited using differential coding which will be dealt with in section 5.14. However, errors in the magnitude of DC coefficients cause wide area flicker in the picture and so the coding must be accurate.

Typical material often has many coefficients which are zero valued, especially after requantizing. The distribution of these also follows a pattern. The non-zero values tend to be found in the top-left-hand corner of the DCT block, but as the radius increases, not only do the coefficient values fall, but it becomes increasingly likely that these small coefficients will be interspersed with zero-valued coefficients. As the radius increases further it is probable that a region where all coefficients are zero will be entered.

MPEG uses all these attributes of DCT coefficients when encoding a coefficient block. By sending the coefficients in an optimum order, by describing their values with Huffman coding and by using run-length encoding for the zero-valued coefficients it is possible to achieve a significant reduction in coefficient data which remains entirely lossless. Despite the complexity of this process, it does contribute to improved picture quality because for a given bit rate lossless coding of the coefficients must be better than requantizing, which is lossy. Of course, for lower bit rates both will be required.

Google Exhibit No. 1005
Page 062

Google Exhibit No. 1063
Page 0317

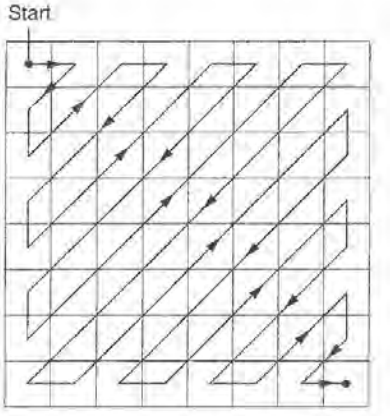


Figure 5.32 The zig-zag scan for a progressively scanned image.

It is an advantage to scan in a sequence where the largest coefficient values are scanned first. Then the next coefficient is more likely to be zero than the previous one. With progressively scanned material, a regular zig-zag scan begins in the top-left corner and ends in the bottom-right corner as shown in Figure 5.32. Zig-zag scanning means that significant values are more likely to be transmitted first, followed by the zero values. Instead of coding these zeros, an unique 'end of block' (EOB) symbol is transmitted instead.

As the zig-zag scan approaches the last finite coefficient it is increasingly likely that some zero value coefficients will be scanned. Instead of transmitting the coefficients as zeros, the *zero-run-length*, i.e. the number of zero-valued coefficients in the scan sequence, is encoded into the next non-zero coefficient which is itself variable-length coded. This combination of run-length and variable-length coding is known as RLC/VLC in MPEG.

The DC coefficient is handled separately because it is differentially coded and this discussion relates to the AC coefficients. Three items need to be handled for each coefficient: the zero-run-length prior to this coefficient, the wordlength and the coefficient value itself. The word-length needs to be known by the decoder so that it can correctly parse the bitstream. The wordlength of the coefficient is expressed directly as an integer called the *size*.

Figure 5.33(a) shows that a two-dimensional run/size table is created. One dimension expresses the zero-run-length; the other the size. A run length of zero is obtained when adjacent coefficients are non-zero, but a code of 0/0 has no meaningful run/size interpretation and so this bit pattern is used for the end-of-block (EOB) symbol.

Google Exhibit No. 1005
Page 063

Google Exhibit No. 1063
Page 0318

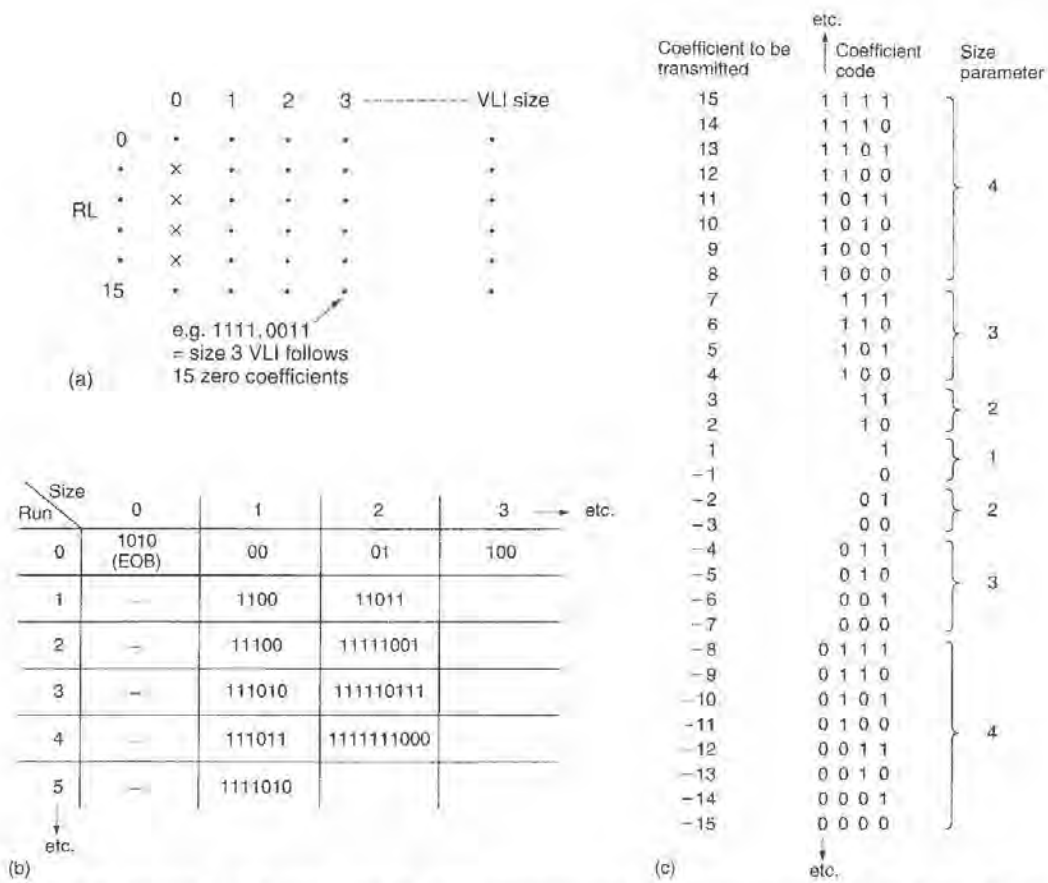


Figure 5.33 Run-length and variable-length coding simultaneously compresses runs of zero-valued coefficients and describes the wordlength of a non-zero coefficient.

In the case where the zero-run-length exceeds 14, a code of 15/0 is used signifying that there are fifteen zero-valued coefficients. This is then followed by another run/size parameter whose run-length value is added to the previous fifteen.

The run/size parameters contain redundancy because some combinations are more common than others. Figure 5.33(b) shows that each run/size value is converted to a variable-length Huffman codeword for transmission. As was shown in section 1.5, the Huffman codes are designed so that short codes are never a prefix of long codes so that the decoder can deduce the parsing by testing an increasing number of bits until a match with the look-up table is found. Having parsed and decoded the Huffman run/size code, the decoder then knows what the coefficient wordlength will be and can correctly parse that.

Google Exhibit No. 1005
Page 064

Google Exhibit No. 1063
Page 0319

The variable-length coefficient code has to describe a bipolar coefficient, i.e one which can be positive or negative. Figure 5.33(c) shows that for a particular size, the coding scale has a certain gap in it. For example, all values from – 7 to + 7 can be sent by a size 3 code, so a size 4 code only has to send the values of – 15 to – 8 and + 8 to + 15. The coefficient code is sent as a pure binary number whose value ranges from all zeros to all ones where the maximum value is a function of the size. The number range is divided into two, the lower half of the codes specifying negative values and the upper half specifying positive.

In the case of positive numbers, the transmitted binary value is the actual coefficient value, whereas in the case of negative numbers a constant must be subtracted which is a function of the size. In the case of a size 4 code, the constant is 15_{10}. Thus a size 4 parameter of 0111_2 (7_{10}) would be interpreted as 7 – 15 = – 8. A size of 5 has a constant of 31 so a transmitted coded of 01010_2 (10_2) would be interpreted as 10 – 31 = –21.

This technique saves a bit because, for example, 63 values from – 31 to + 31 are coded with only five bits having only 32 combinations. This is possible because that extra bit is effectively encoded into the run/size parameter.

Figure 5.34 shows the entire MPEG-1 and MPEG-2 spatial coding subsystem. Macroblocks are subdivided into DCT blocks and the DCT is calculated. The resulting coefficients are multiplied by the weighting matrix and then requantized. The coefficients are then reordered by the zig-zag scan so that full advantage can be taken of run-length and variable-length coding. The last non-zero coefficient in the scan is followed by the EOB symbol.

In predictive coding, sometimes the motion-compensated prediction is nearly exact and so the prediction error will be almost zero. This can also happen on still parts of the scene. MPEG takes advantage of this by sending a code to tell the decoder there is no prediction error data for the macroblock concerned.

The success of temporal coding depends on the accuracy of the vectors. Trying to reduce the bit rate by reducing the accuracy of the vectors is false economy as this simply increases the prediction error. Consequently for a given GOP structure it is only in the the spatial coding that the

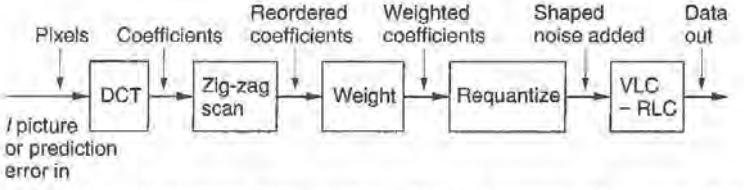


Figure 5.34 A complete spatial coding system which can compress an *I* picture or the prediction error in *P* and *B* pictures. See text for details.

Google Exhibit No. 1005
Page 065

Google Exhibit No. 1063
Page 0320

overall bit rate is determined. The RLC/VLC coding is lossless and so its contribution to the compression cannot be varied. If the bit rate is too high, the only option is to increase the size of the coefficient-requantizing steps. This has the effect of shortening the wordlength of large coefficients, and rounding small coefficients to zero, so that the bit rate goes down. Clearly if taken too far the picture quality will also suffer because at some point the noise floor will become visible as some form of artifact.

5.13 A bidirectional coder

MPEG does not specify how an encoder is to be built or what coding decisions it should make. Instead it specifies the protocol of the bitstream at the output. As a result the coder shown in Figure 5.35 is only an example. Conceptually MPEG-2 Main Profile coding with a progressive scan input is almost identical to that of MPEG-1 as the differences are to be found primarily in the picture size and the resulting bit rate. The coder shown here will create MPEG-1 and MPEG-2 bidirectionally coded bitstreams from progressively scanned inputs. It also forms the basis of interlaced MPEG-2 coding and for MPEG-4 texture coding, both of which will be considered later.

Figure 5.35(a) shows the component parts of the coder. At the input is a chain of picture stores which can be bypassed for reordering purposes. This allows a picture to be encoded ahead of its normal timing when bidirectional coding is employed.

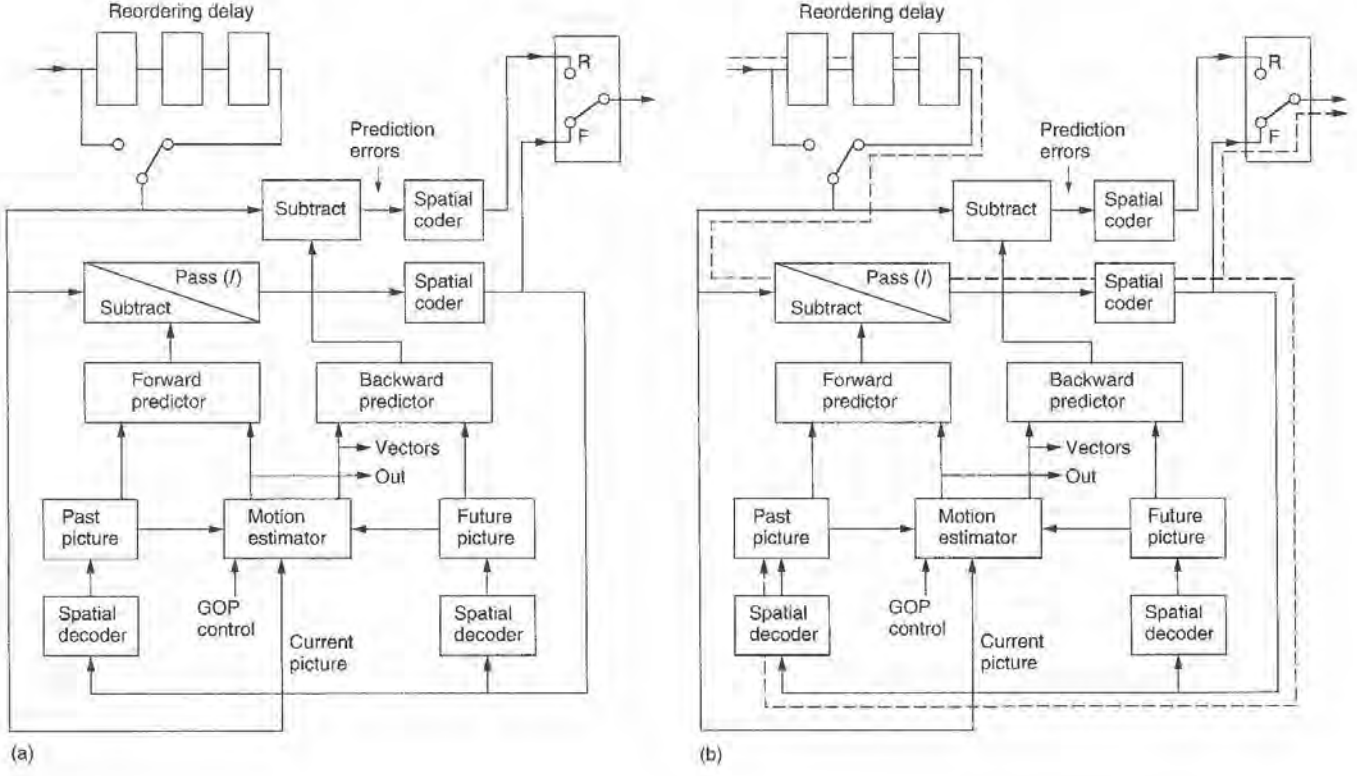
At the centre is a dual-motion estimator which can simultaneously measure motion between the input picture, an earlier picture and a later picture. These reference pictures are held in frame stores. The vectors from the motion estimator are used locally to shift a picture in a frame store to form a predicted picture. This is subtracted from the input picture to produce a prediction error picture which is then spatially coded.

The bidirectional encoding process will now be described. A GOP begins with an *I* picture which is intra-coded. In Figure 5.35(b) the *I* picture emerges from the reordering delay. No prediction is possible on an *I* picture so the motion estimator is inactive. There is no predicted picture and so the prediction error subtractor is set simply to pass the input. The only processing which is active is the forward spatial coder which describes the picture with DCT coefficients. The output of the forward spatial coder is locally decoded and stored in the past picture frame store.

The reason for the spatial encode/decode is that the past picture frame store now contains exactly what the decoder frame store will contain, including the effects of any requantizing errors. When the same picture is

Google Exhibit No. 1005
Page 066

Google Exhibit No. 1063
Page 0321



Google Exhibit No. 1005
Page 067

Google Exhibit No. 1063
Page 0322

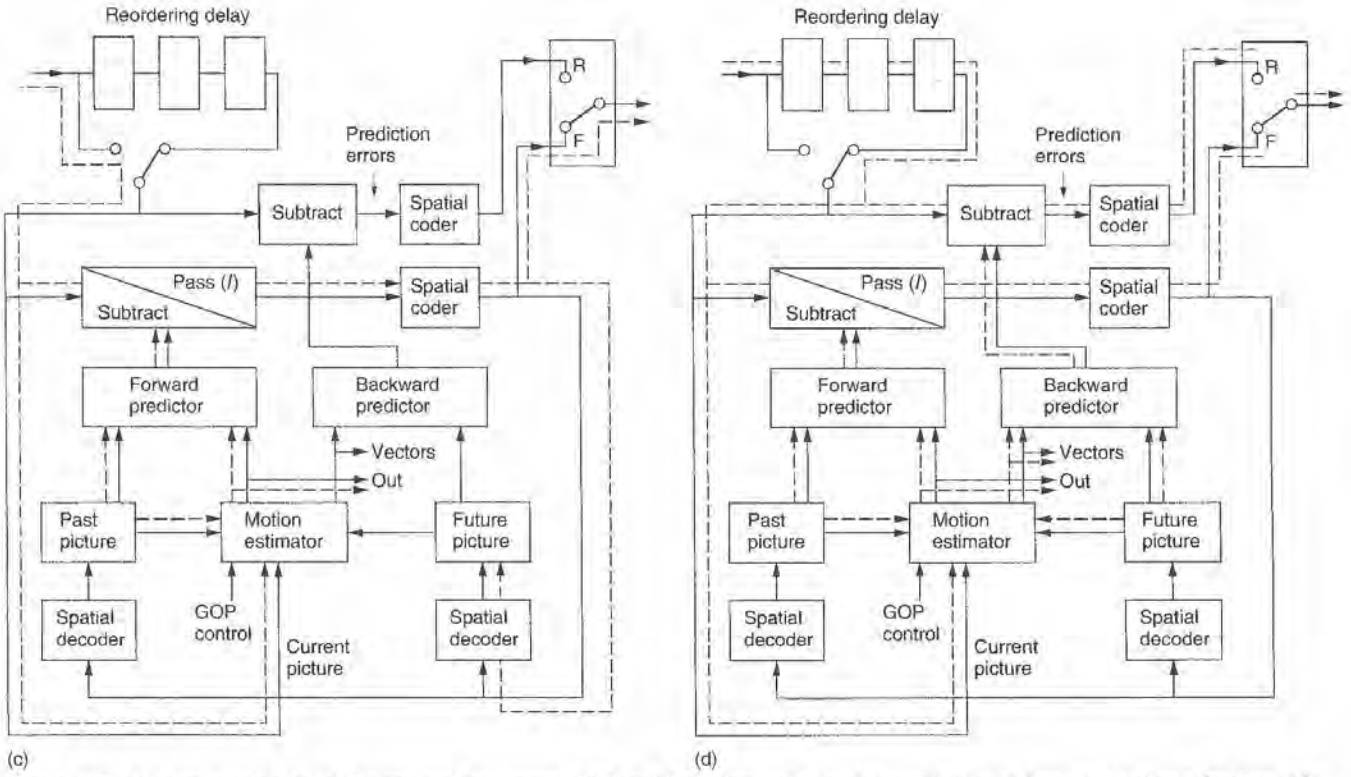


Figure 5.35 A bidirectional coder. (a) The essential components. (b) Signal flow when coding an *I* picture. (c) Signal flow when coding a *P* picture. (d) Signal flow when bidirectional coding.

Google Exhibit No. 1005
Page 068

Google Exhibit No. 1063
Page 0323

used as a reference at both ends of a differential coding system, the errors will cancel out.

Having encoded the *I* picture, attention turns to the *P* picture. The input sequence is *IBBP*, but the transmitted sequence must be *IPBB*. Figure 5.35(c) shows that the reordering delay is bypassed to select the *P* picture. This passes to the motion estimator which compares it with the *I* picture and outputs a vector for each macroblock. The forward predictor uses these vectors to shift the *I* picture so that it more closely resembles the *P* picture. The predicted picture is then subtracted from the actual picture to produce a forward prediction error. This is then spatially coded. Thus the *P* picture is transmitted as a set of vectors and a prediction error image.

The *P* picture is locally decoded in the right-hand decoder. This takes the forward-predicted picture and adds the decoded prediction error to obtain exactly what the decoder will obtain.

Figure 5.35(d) shows that the encoder now contains an *I* picture in the left store and a *P* picture in the right store. The reordering delay is reselected so that the first *B* picture can be input. This passes to the motion estimator where it is compared with both the *I* and *P* pictures to produce forward and backward vectors. The forward vectors go to the forward predictor to make a *B* prediction from the *I* picture. The backward vectors go to the backward predictor to make a *B* prediction from the *P* picture. These predictions are simultaneously subtracted from the actual *B* picture to produce a forward prediction error and a backward prediction error. These are then spatially encoded. The encoder can then decide which direction of coding resulted in the best prediction; i.e. the smallest prediction error.

In the encoder shown, motion estimation sometimes takes place between an input picture and a decoded picture. With increased complexity, the original input pictures may be stored for motion-estimation purposes. As these contain no compression artifacts the motion vectors may be more accurate.

Not shown in the interests of clarity is a third signal path which creates a predicted *B* picture from the average of forward and backward predictions. This is subtracted from the input picture to produce a third prediction error. In some circumstances this prediction error may use fewer data than either forward of backward prediction alone.

As *B* pictures are never used to create other pictures, the decoder does not locally decode the *B* picture. After decoding and displaying the *B* picture the decoder will discard it. At the encoder the *I* and *P* pictures remain in their frame stores and the second *B* picture is input from the reordering delay.

Following the encoding of the second *B* picture, the encoder must reorder again to encode the second *P* picture in the GOP. This will be

Google Exhibit No. 1005
Page 069

Google Exhibit No. 1063
Page 0324

locally decoded and will replace the *I* picture in the left store. The stores and predictors switch designation because the left store is now a future *P* picture and the right store is now a past *P* picture. *B* pictures between them are encoded as before.

5.14 Slices

In MPEG-1 and MPEG-2 *I* pictures and in MPEG-4 *I-VOPs*, the DC coefficient describes the average brightness of an entire DCT block. In real video the DC component of adjacent blocks will be similar much of the time. A saving in bit rate can be obtained by differentially coding the DC coefficient.

In *P* and *B* pictures this is not done because these are prediction errors, not actual images, and the statistics are different. However, *P* and *B* pictures send vectors and instead the redundancy in these is explored. In a large moving object, many macroblocks will be moving at the same velocity and their vectors will be the same. Thus differential vector coding will be advantageous.

As has been seen above, differential coding cannot be used indiscriminately as it is prone to error propagation. Periodically absolute DC coefficients and vectors must be sent and the *slice* is the logical structure which supports this mechanism. In *I* coding, the first DC coefficient in a slice is sent in absolute form, whereas the subsequent coefficients are sent differentially. In *P* or *B* coding, the first vector in a slice is sent in absolute form, but the subsequent vectors are differential.

Slices are horizontal picture strips which are one macroblock (16 pixels) high and which proceed from left to right across the screen and may run on to the next horizontal row. The encoder is free to decide how big slices should be and where they begin and end.

In the case of a central dark building silhouetted against the bright sky, there would be two large changes in the DC coefficients, one at each edge of the building. It may be advantageous to the encoder to break the width of the picture into three slices, one each for the left and right areas of sky and one for the building. In the case of a large moving object, different slices may be used for the object and the background.

Each slice contains its own synchronizing pattern, so following a transmission error, correct decoding can resume at the next slice. Slice size can also be matched to the characteristics of the transmission channel. For example, in an error-free transmission system the use of a large number of slices in a packet simply wastes data capacity on surplus synchronizing patterns. However, in a non-ideal system it might be advantageous to have frequent resynchronizing. In DVB, for example, additional constraints are placd on the slice size.

Google Exhibit No. 1005
Page 070

Google Exhibit No. 1063
Page 0325

chroma macroblock of each component type. As it cannot be divided in half, this chroma is deemed to belong to the luminance DCT blocks of the upper field. In 4:2:2 no such difficulty arises.

MPEG supports interlace simply because interlaced video exists in legacy systems and there is a requirement to compress it. However, where the opportunity arises to define a new system, interlace should be avoided. Legacy interlaced source material should be handled using a motion-compensated de-interlacer prior to compression in the progressive domain.

5.16 MPEG-1 and MPEG-2 coders

Figure 5.40 shows a complete coder. The bidirectional coder outputs coefficients and vectors, and the quantizing table in use. The vectors of *P* and *B* pictures and the DC coefficients of *I* pictures are differentially encoded in slices and the remaining coefficients are RLC/VLC coded. The multiplexer assembles all these data into a single bitstream called an elementary stream. The output of the encoder is a buffer which absorbs the variations in bit rate between different picture types. The buffer output has a constant bit rate determined by the demand clock. This comes from the transmission channel or storage device. If the bit rate is low, the buffer will tend to fill up, whereas if it is high the buffer will tend to empty. The buffer content is used to control the severity of the requantizing in the spatial coders. The more the buffer fills, the bigger the requantizing steps get.

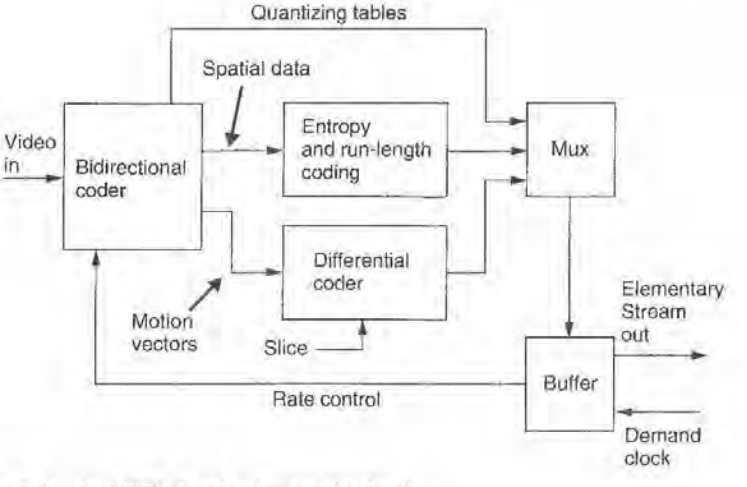


Figure 5.40 An MPEG-2 coder. See text for details.

Google Exhibit No. 1005
Page 071

Google Exhibit No. 1063
Page 0326

The buffer in the decoder has a finite capacity and the encoder must model the decoder's buffer occupancy so that it neither overflows nor underflows. An overflow might occur if an *I* picture is transmitted when the buffer content is already high. The buffer occupancy of the decoder depends somewhat on the memory access strategy of the decoder. Instead of defining a specific buffer size, MPEG-2 defines the size of a particular mathematical model of a hypothetical buffer. The decoder designer can use any strategy which implements the model, and the encoder can use any strategy which doesn't overflow or underflow the model. The elementary stream has a parameter called the video buffer verifier (VBV) which defines the minimum buffering assumptions of the encoder.

As was seen in Chapter 1, buffering is one way of ensuring constant quality when picture entropy varies. An intelligent coder may run down the buffer contents in anticipation of a difficult picture sequence so that a large amount of data can be sent.

MPEG-2 does not define what a decoder should do if a buffer underflow or overflow occurs, but since both irrecoverably lose data it is obvious that there will be more or less of an interruption to the decoding. Even a small loss of data may cause loss of synchronization and in the case of long GOP the lost data may make the rest of the GOP undecodable. A decoder may chose to repeat the last properly decoded picture until it can begin to operate correctly again.

Buffer problems occur if the VBV model is violated. If this happens then more than one underflow or overflow can result from a single violation. Switching an MPEG bitstream can cause a violation because the two encoders concerned may have radically different buffer occupancy at the switch.

5.17 The elementary stream

Figure 5.41 shows the structure of the elementary stream from an MPEG-2 encoder. The structure begins with a set of coefficients representing a DCT block. Six or eight DCT blocks form the luminance and chroma content of one macroblock. In *P* and *B* pictures a macroblock will be associated with a vector for motion compensation. Macroblocks are associated into slices in which DC coefficients of *I* pictures and vectors in *P* and *B* pictures are differentially coded. An arbitrary number of slices forms a picture and this needs *I*/*P*/*B* flags describing the type of picture it is. The picture may also have a global vector which efficiently deals with pans.

Several pictures form a group of pictures (GOP). The GOP begins with an *I* picture and may or may not include *P* and *B* pictures in a structure which may vary dynamically.

Google Exhibit No. 1005
Page 072

Google Exhibit No. 1063
Page 0327

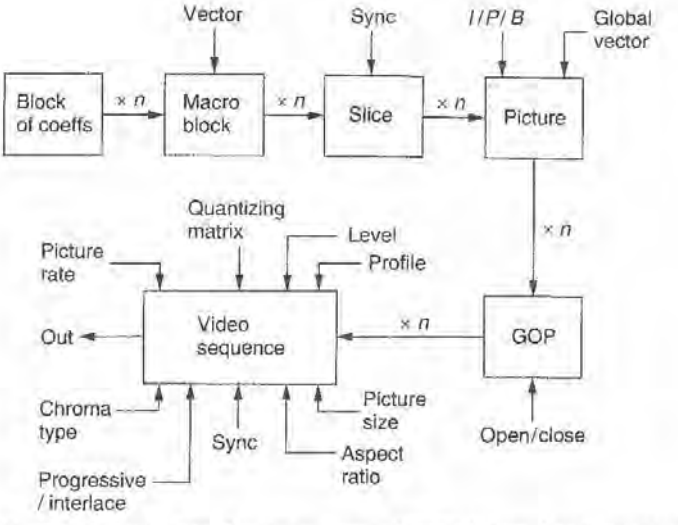


Figure 5.41 The structure of an elementary stream. MPEG defines the syntax precisely.

Several GOPs form a sequence which begins with a sequence header containing important data to help the decoder. It is possible to repeat the header within a sequence, and this helps lock-up in random access applications. The sequence header describes the MPEG-2 profile and level, whether the video is progressive or interlaced, whether the chroma is 4:2:0 or 4:2:2, the size of the picture and the aspect ratio of the pixels. The quantizing matrix used in the spatial coder can also be sent. The sequence begins with a standardized bit pattern which is detected by a decoder to synchronize the deserialization.

5.18 An MPEG-2 decoder

The decoder is only defined by implication from the definitions of syntax and any decoder which can correctly interpret all combinations of syntax at a particular profile will be deemed compliant however it works. Compliant MPEG-2 decoders must be able to decode an MPEG-1 elementary stream. This is not particularly difficult as MPEG-2 uses all the tools of MPEG-1 and simply adds more of its own. Consequently an MPEG-1 bitstream can be thought of as a subset of an MPEG-2 bitstream.

The first problem a decoder has is that the input is an endless bitstream which contains a huge range of parameters many of which have variable length. Unique synchronizing patterns must be placed periodically throughout the bitstream so that the decoder can identify known starting points. The pictures which can be sent under MPEG are so flexible that

Google Exhibit No. 1005
Page 073

Google Exhibit No. 1063
Page 0328

the decoder must first find a sequence header so that it can establish the size of the picture, the frame rate, the colour coding used, etc.

The decoder must also be supplied with a 27 MHz system clock. In a DVD player, this would come from a crystal, but in a transmission system this would be provided by a numerically locked loop running from a clock reference parameter in the bitstream (see Chapter 6). Until this loop has achieved lock the decoder cannot function properly.

Figure 5.42 shows a bidirectional decoder. The decoder can only begin decoding with an *I* picture and as this only uses intra-coding there will be no vectors. An *I* picture is transmitted as a series of slices which begin with subsidiary synchronizing patterns. The first macroblock in the slice contains an absolute DC coefficient, but the remaining macroblocks code the DC coefficient differentially so the decoder must subtract the differential values from the previous value to obtain the absolute value.

The AC coefficients are sent as Huffman coded run/size parameters followed by coefficient value codes. The variable-length Huffman codes are decoded by using a look-up table and extending the number of bits considered until a match is obtained. This allows the zero-run-length and the coefficient size to be established. The right number of bits is taken from the bitstream corresponding to the coefficient code and this is decoded to the actual coefficient using the size parameter.

If the correct number of bits has been taken from the stream, the next bit must be the beginning of the next run/size code and so on until the EOB symbol is reached. The decoder uses the coefficient values and the zero-run-lengths to populate a DCT coefficient block following the appropriate zig-zag scanning sequence. Following EOB, the bitstream then continues with the next DCT block. Clearly this Huffman decoding will work perfectly or not at all. A single bit slippage in synchronism or a single corrupted data bit can cause a spectacular failure.

Once a complete DCT coefficient block has been received, the coefficients need to be inverse quantized and inverse weighted. Then an inverse DCT can be performed and this will result in an 8×8 pixel block. A series of DCT blocks will allow the luminance and colour information for an entire macroblock to be decoded and this can be placed in a frame store. Decoding continues in this way until the end of the slice when an absolute DC coefficient will once again be sent. Once all the slices have been decoded, an entire picture will be resident in the frame store.

The amount of data needed to decode the picture is variable and the decoder just keeps going until the last macroblock is found. It will obtain data from the input buffer. In a constant bit rate transmission system, the decoder will remove more data to decode an *I* picture than has been received in one picture period, leaving the buffer emptier than it began. Subsequent *P* and *B* pictures need much fewer data and allow the buffer to fill again.

Google Exhibit No. 1005
Page 074

Google Exhibit No. 1063
Page 0329

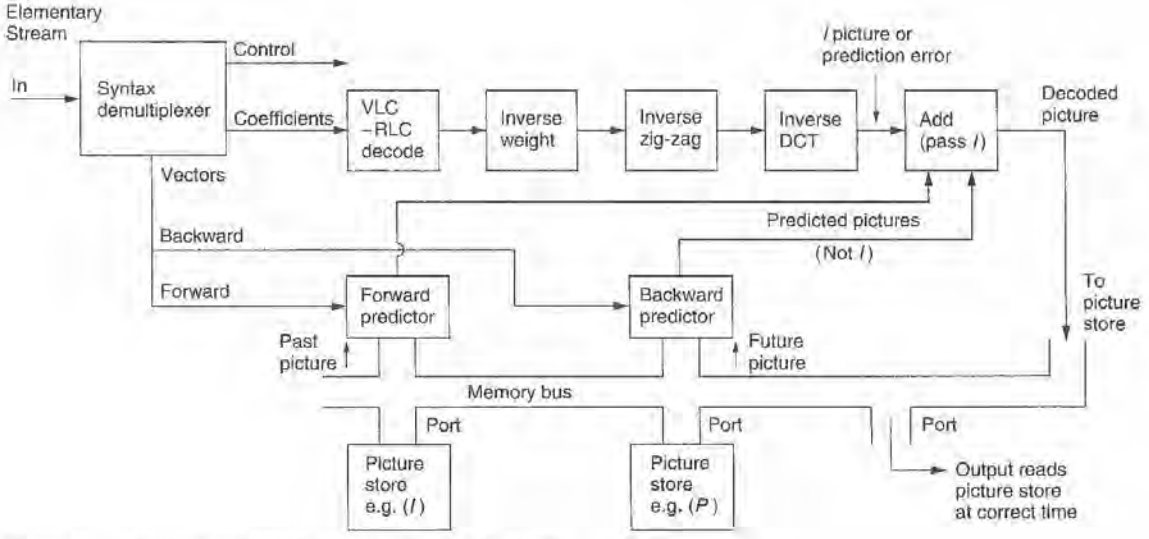


Figure 5.42 A bidirectional MPEG-2 decoder. See text for details.

Google Exhibit No. 1005
Page 075

Google Exhibit No. 1063
Page 0330

The picture will be output when the time stamp (see Chapter 6) sent with the picture matches the state of the decoder's time count.

Following the *I* picture may be another *I* picture or a *P* picture. Assuming a *P* picture, this will be predictively coded from the *I* picture. The *P* picture will be divided into slices as before. The first vector in a slice is absolute, but subsequent vectors are sent differentially. However, the DC coefficients are not differential.

Each macroblock may contain a forward vector. The decoder uses this to shift pixels from the *I* picture into the correct position for the predicted *P* picture. The vectors have half-pixel resolution and where a half-pixel shift is required, an interpolator will be used.

The DCT data is sent much as for an *I* picture. It will require inverse quantizing, but not inverse weighting because *P* and *B* coefficients are flat-weighted. When decoded this represents an error-cancelling picture which is added pixel-by-pixel to the motion-predicted picture. This results in the output picture.

If bidirectional coding is being used, the *P* picture may be stored until one or more *B* pictures have been decoded. The *B* pictures are sent essentially as a *P* picture might be, except that the vectors can be forward, backward or bidirectional. The decoder must take pixels from the *I* picture, the *P* picture, or both, and shift them according to the vectors to make a predicted picture. The DCT data decodes to produce an error-cancelling image as before.

In an interlaced system, the prediction mechanism may alternatively obtain pixel data from the previous field or the field before that. Vectors may relate to macroblocks or to 16 × 8 pixel areas. DCT blocks after decoding may represent frame lines or field lines. This adds up to a lot of different possibilities for a decoder handling an interlaced input.

5.19 MPEG-4

As was seen in Chapter 1, MPEG-4 advances the coding art in a number of ways. Whereas MPEG-1 and MPEG-2 were directed only to coding the video pictures which resulted after shooting natural scenes or from computer synthesis, MPEG-4 also moves further back in the process of how those scenes were created. For example, the rotation of a detailed three-dimensional object before a video camera produces huge changes in the video from picture to picture which MPEG-2 would find difficult to code. Instead, if the three-dimensional object is re-created at the decoder, rotation can be portrayed by transmitting a trivially small amount of vector data.

If the above object is synthetic, effectively the synthesis or rendering process is completed in the decoder. However, a suitable if complex

Google Exhibit No. 1005
Page 076

Google Exhibit No. 1063
Page 0331

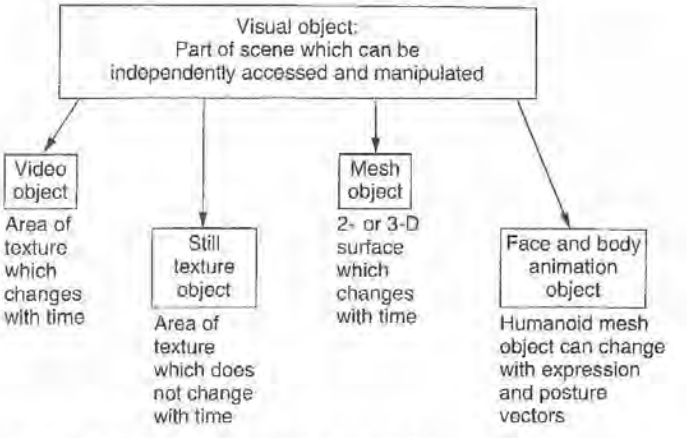


Figure 5.43 In MPEG-4 four types of *objects* are coded.

image processor at the encoder could identify such objects in natural scenes. MPEG-4 objects are defined as a part of a scene which can independently be accessed or manipulated. An object is an entity that exists over a certain time span. The pictures of conventional imaging become *object planes* in MPEG-4. Where an object intersects an object plane, it can be described by the coding system using intra-coding, forward prediction or bidirectional prediction.

Figure 5.43 shows that MPEG-4 has four object types. A video object is an arbitrarily shaped planar pixel array describing the appearance or *texture* of part of a scene. A still texture object or *sprite* is a planar video object in which there is no change with respect to time. A mesh object describes a two- or three-dimensional shape as a set of points. The shape and its position can change with respect to time. Using computer graphics techniques, texture can be mapped onto meshes, a process known as warping, to produce rendered images.

Using two-dimensional warping, a still texture object can be made to move. In three-dimensional graphic rendering, mesh coding allows an arbitrary solid shape to be created which is then covered with texture. Perspective computation then allows this three-dimensional object to be viewed in correct perspective from any viewpoint. MPEG-4 provides tools to allow two- or three-dimensional meshes to be created in the decoder and then oriented by vectors. Changing the vectors then allows realistic moving images to be created with an extremely low bit rate.

Face and body animation is a specialized subset of three-dimensional mesh coding in which the mesh represents a human face and/or body. As the subject moves, carefully defined vectors carry changes of expression which allow rendering of an apparently moving face and/or body which has been almost entirely synthesized from a single still picture.

Google Exhibit No. 1005
Page 077

Google Exhibit No. 1063
Page 0332

In addition to object coding, MPEG-4 refines the existing MPEG tools by increasing the efficiency of a number of processes using lossless prediction. This improves the performance of the motion compensation and coefficient coding allowing either a lower bit rate or improved quality. MPEG-4 also extends the idea of scaleability introduced in MPEG-2. Multiple scaleability is supported, where a low-bit-rate base-level picture may optionally be enhanced by adding information from one or more additional bitstreams. This approach is useful in network applications where the content creator cannot know the bandwidth which a particular user will have available. Scaleability allows the best quality in the available bandwidth.

Although most of the spatial compression of MPEG-4 is based on the DCT as in earlier MPEG standards, MPEG-4 also introduces wavelet coding of still objects. Wavelets are advantageous in scaleable systems because they naturally decompose the original image into various resolutions.

5.20 Video objects

Figure 5.44 shows an example of a video object intersecting *video object planes*, or VOPs. At each plane, the shape and the texture of the object must be portrayed. Figure 5.45 shows this can be done using appropriate

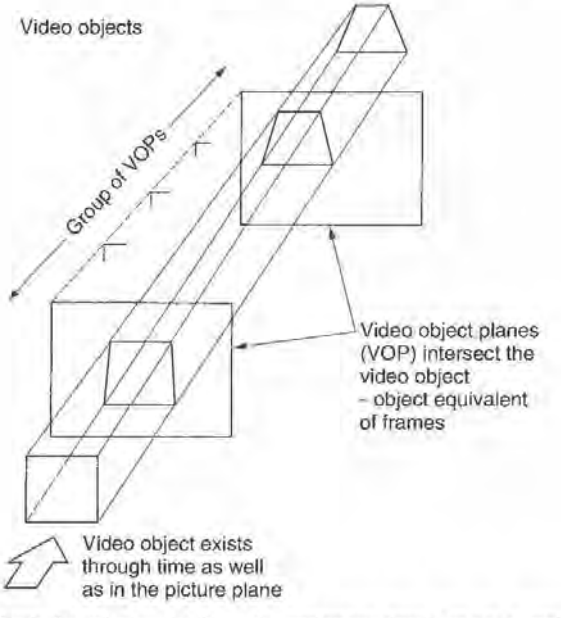


Figure 5.44 Effectively objects exist through continuous time, but are sampled for display purposes. A video object plane (VOP) is a temporal sample of an object. VOPs correspond to pictures of MPEG-1 and MPEG-2.

Google Exhibit No. 1005
Page 078

Google Exhibit No. 1063
Page 0333

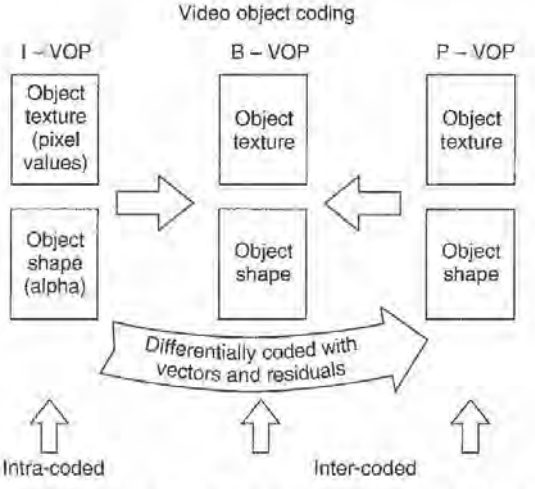


Figure 5.45 VOPs can be encoded using the same intra- and inter-coding approach of MPEG-1 and MPEG-2.

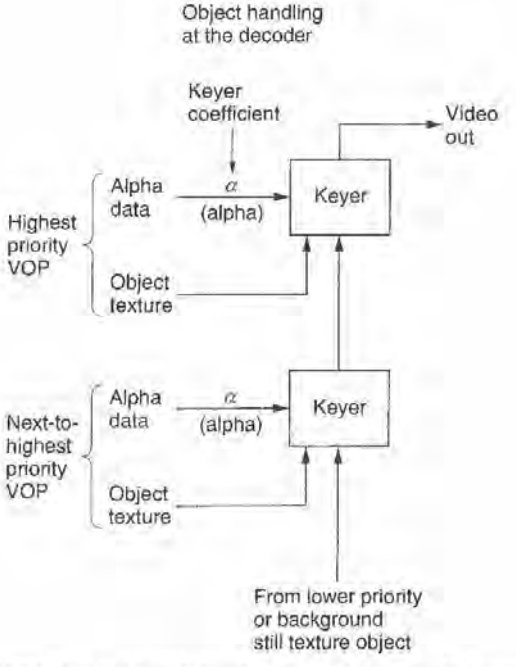


Figure 5.46 An MPEG-4 object-based decoder is essentially a multi-layered compositing device, using keying to give the illusion that objects are in front of a background.

Google Exhibit No. 1005
Page 079

Google Exhibit No. 1063
Page 0334

combinations of intra- and inter-coding as described for the earlier standards. This gives rise to I-VOPs, P-VOPs and B-VOPs. A group of VOPs is known as a GOV.

Figure 5.46 shows how video objects are handled at the decoder which effectively contains a multi-layer compositing stage. The shape of each object is transmitted using *alpha data* which is decoded to produce a key signal. This has the effect of keying the texture data into the overall picture. Several objects can be keyed into the same picture. The background may be yet another video object or a still texture object which can be shifted by vectors in the case of pans etc.

Figure 5.47 shows that MPEG-4 codes each arbitrarily shaped video object by creating a *bounding rectangle* within which the object resides. The shape and the texture of each video object are re-created at the decoder by two distinct but interrelated processes. The bounding rectangle exists on macroblock boundaries and can change from VOP to VOP. At each VOP,

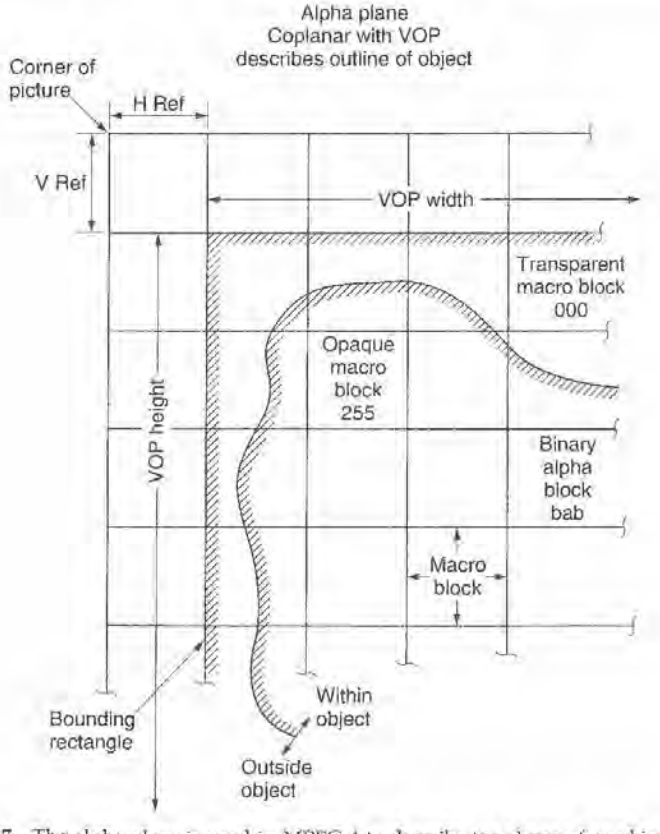


Figure 5.47 The alpha plane is used in MPEG-4 to describe the shape of an object. Within a bounding rectangle are three different types of macroblock: those within the object, those outside and those which are intersected by the boundary.

Google Exhibit No. 1005
Page 080

Google Exhibit No. 1063
Page 0335

the bitstream contains horizontal and vertical references specifying the position of the top-left corner of the bounding rectangle and its width and height.

Within the bounding rectangle are three kinds of macroblock. *Transparent macroblocks* are entirely outside the object and contain no texture data. The shape data are identical over the entire block and result in a key signal which deselects this block in the compositing process. Such shape data are trivial to compress. *Opaque macroblocks* are entirely within the object and are full of texture data. The shape data are again identical over the entire block and result in the block being keyed at the compositor. *Boundary macroblocks* are blocks through which the edge of the object passes. They contain essentially all the shape information and somewhat fewer texture data than an opaque macroblock.

5.21 Texture coding

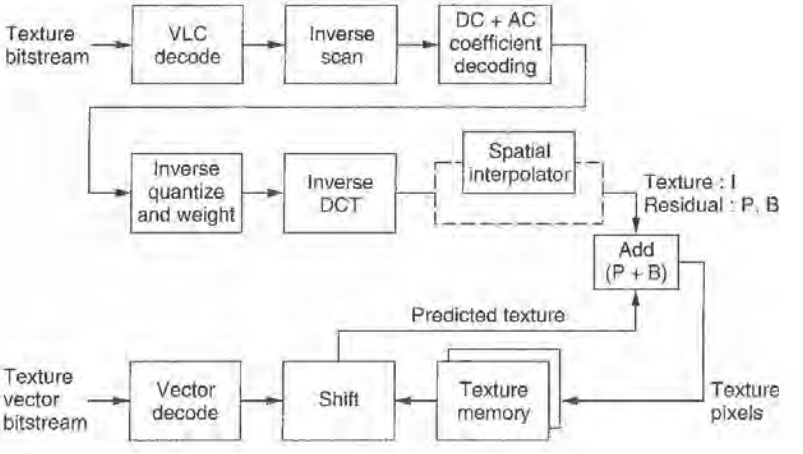
In MPEG-1 and MPEG-2 the only way of representing an image is with pixels and this requires no name. In MPEG-4 there are various types of image description tools and it becomes necessary to give the pixel representation of the earlier standards a name. This is *texture coding* which is that part of MPEG-4 that operates on pixel-based areas of image. Coming later than MPEG-1 and MPEG-2, the MPEG-4 texture-coding system can afford additional complexity in the search for higher performance. Figure 5.48(a) shows the texture decoding system of MPEG-4 and in (b) is a comparison with MPEG-2 showing that it is a refinement of the earlier technique. These refinements are lossless in that the reduction in bit rate they allow does not result in a loss of quality.

When inter-coding, MPEG-4 may use motion-compensated bidirectional coding as for MPEG-2, but in doing so it also uses a more advanced form of lossless vector compression which may then be used to increase the vector density to one vector per DCT block. In inter-coded pictures the prediction of the picture is improved so that the residual to be coded is smaller.

When intra-coding, MPEG-4 looks for further redundancy between coefficients using prediction. When a given DCT block is to be intra-coded, certain of its coefficients will be predicted from adjacent blocks. The choice of the most appropriate block is made by measuring the picture gradient, defined as the rate of change of the DC coefficient. Figure 5.49(a) shows that the three adjacent blocks, A, B and C, are analysed to decide whether to predict from the DCT block above (vertical prediction) or to the left (horizontal prediction). Figure 5.49(b) shows that in vertical prediction the top row of coefficients is predicted from the block above so that only the differences between them need to be coded.

Google Exhibit No. 1005
Page 081

Google Exhibit No. 1063
Page 0336



(a)

MPEG-2 Picture coding	MPEG-4 Texture coding
DCT	DCT
DC coefficient prediction in slice	DC coefficient prediction
	AC coefficient prediction
Single coefficient scan	3 × coefficient scans
Fixed resolution weighting	Reduced resolution weighting
VLC/RLC	VLC/RLC
	Result : more efficient I coding
One vector/macroblock	4 × vectors/macroblock option
Differential vectors in slice	Vector prediction
	Result: more efficient P, B coding

(b)

Figure 5.48 (a) The texture decoding system of MPEG-4. (b) A comparison of MPEG-2 and MPEG-4 shows that the latter is more refined.

Figure 5.49(c) shows that in horizontal prediction the left column of coefficients is predicted from the block on the left so that again only the differences need be coded.

Choosing the blocks above and to the left is important because these blocks will already be available in both the encoder and decoder. By making the same picture gradient measurement, the decoder can establish whether vertical or horizontal prediction has been used and so no flag is needed in the bitstream.

Some extra steps are needed to handle the top row and the left column of a picture or object where true prediction is impossible. In these cases both encoder and decoder assume standardized constant values for the missing prediction coefficients.

The picture gradient measurement determines the direction in which there is the least change from block to block. There will generally be fewer DCT coefficients present in this direction. There will be more coefficients in the other axis where there is more change. Consequently it is

Google Exhibit No. 1005
Page 082

Google Exhibit No. 1063
Page 0337

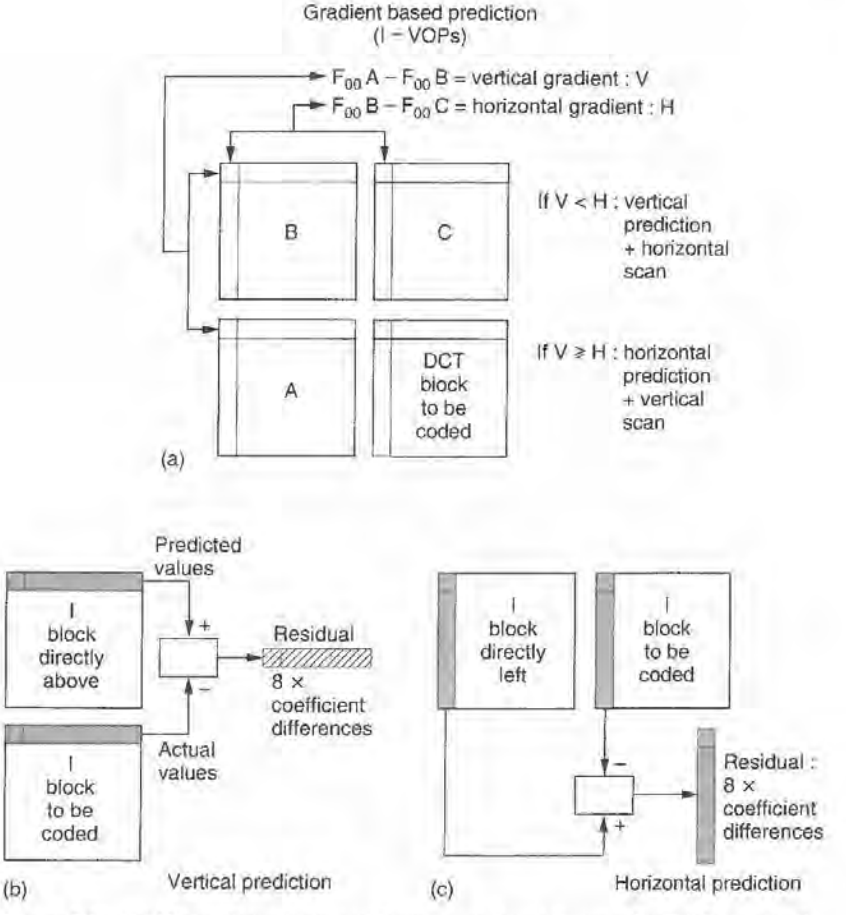


Figure 5.49 (a) DC coefficients are used to measure the picture gradient. (b) In vertical prediction the top row of coefficients are predicted using those above as a basis. (c) In horizontal prediction the left column of coefficients are predicted from those to the left.

advantageous to alter the scanning sequence so that the coefficients which are likely to exist are transmitted earlier in the sequence.

Figure 5.50 shows the two alternate scans for MPEG-4. The alternate horizontal scan concentrates on horizontal coefficients early in the scan and will be used in conjunction with vertical prediction. Conversely the alternate vertical scan concentrates on vertical coefficients early in the scan and will be used in conjunction with horizontal prediction. The decoder can establish which scan has been used in the encoder from the picture gradient.

Coefficient prediction is not employed when inter-coding because the statistics of residual images are different. Instead of attempting to predict residual coefficients, in inter-coded texture, pixel-based prediction may

Google Exhibit No. 1005
Page 083

Google Exhibit No. 1063
Page 0338

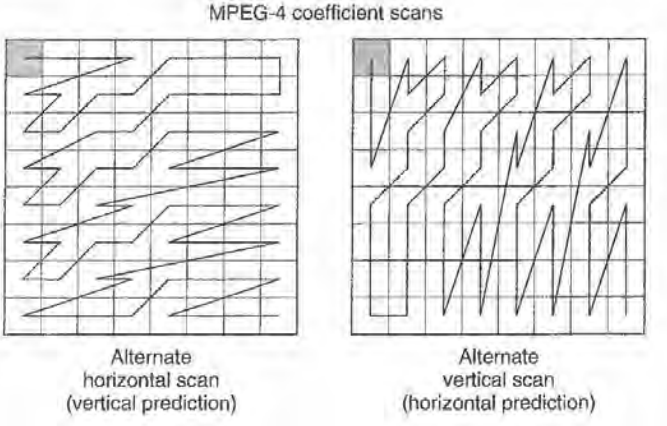


Figure 5.50 The alternate zig-zag scans employed with vertical or horizontal prediction.

be used to reduce the magnitude of texture residuals. This technique is known as *overlapped block motion compensation* (OBMC) which is only used in P-VOPs. With only one vector per DCT block, clearly in many cases the vector cannot apply to every pixel in the block. If the vector is considered to describe the motion of the centre of the block, the vector accuracy falls towards the edge of the block. A pixel in the corner of a block is almost equidistant from a vector in the centre of an adjacent block.

OBMC uses vectors from adjacent blocks, known as *remote vectors* in addition to the vector of the current block for prediction. Figure 5.51 shows that the motion-compensation process of MPEG-1 and MPEG-2 which uses a single vector is modified by the addition of the pixel prediction system which considers three vectors. A given pixel in the block to be coded is predicted from the weighted sum of three motion-compensated pixels taken from the previous I- or P-VOP. One of these pixels is obtained in the normal way by accessing the previous VOP with a shift given by the vector of this block. The other two are obtained by accessing the same VOP pixels using the remote vectors of two adjacent blocks.

The remote vectors which are used and the weighting factors are both a function of the pixel position in the block. Figure 5.51 shows that the block to be coded is divided into quadrants. The remote vectors are selected from the blocks closest to the quadrant in which the pixel resides. For example, a pixel in the bottom-right quadrant would be predicted using remote vectors from the DCT block immediately below and the block immediately to the right.

Not all blocks can be coded in this way. In P-VOPs it is permissible to have blocks which are not coded or intra-blocks which contain no vector. Remote vectors will not all be available at the boundaries of a VOP. In the

Google Exhibit No. 1005
Page 084

Google Exhibit No. 1063
Page 0339

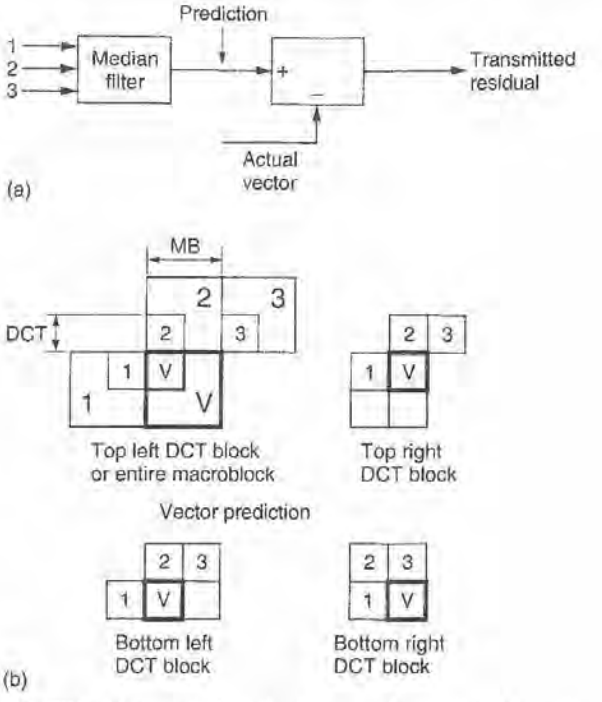


Figure 5.51 (a) MPEG-4 inter-coded pixel values may be predicted from three remote vectors as well as the true vector. (b) The three remote vectors are selected according to the macroblock quadrant. Macroblocks may have one or four vectors and the prediction mechanism allows prediction between blocks of either type.

normal sequence of macroblock transmission, vectors from macroblocks below the current block are not yet available. Some additional steps are needed to handle these conditions. Adjacent to boundaries where a remote vector is not available it is replaced by a copy of the actual vector. This is also done where an adjacent block is intra-coded and for blocks at the bottom of a macroblock where the vectors for the macroblocks below will not be available yet. In the case of non-coded blocks the remote vector is set to zero.

Figure 5.52(a) shows that the weighting factors for pixels near the centre of a block favour that block. In the case of pixels at the corner of the block, the weighting is even between the value obtained from the true vector and the sum of the two pixel values obtained from remote vectors.

The weighted sum produces a predicted pixel which is subtracted from the actual pixel in the current VOP to be coded to produce a residual pixel. Blocks of residual pixels are DCT coded as usual. OBMC reduces the magnitude of residual pixels and gives a corrsponding reduction in the number or magnitude of DCT coefficients to be coded.

Google Exhibit No. 1005
Page 085

Google Exhibit No. 1063
Page 0340

4	5	5	5	5	5	5	4
5	5	5	5	5	5	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4

(a) Weighting values for use with motion vector of current Y block (true vector)

2	2	2	2	2	2	2	2
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
2	2	2	2	2	2	2	2

(b) Weighting values for use with motion vector of Y blocks above or below current Y block (remote vector)

2	1	1	1	1	1	1	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	1	1	1	1	1	1	2

(c) Weighting values for use with motion vectors of Y blocks left or right of current Y block (remote vector)

Figure 5.52 Vector prediction.

OBMC is lossless because the decoder already has access to all the vectors and knows the weighting tables. Consequently the only overhead is the transmission of a flag which enables or disables the mechanism.

MPEG-4 also has the ability to downsample prediction error or residual macroblocks which contain little detail. A 16 × 16 macroblock block is downsampled to 8 × 8 and flagged. The decoder will identify the flag and interpolate back to 16 × 16.

In vector prediction, each macroblock may have only one or four vectors as the coder decides. Consequently the prediction of a current vector may have to be done from either macroblock or DCT block vectors. In the case of predicting one vector for an entire macroblock, or the top-left DCT block vector, the process shown in Figure 5.51(b) is used. Three earlier vectors, which may be macroblock or DCT block vectors, as available, are used as the input to the prediction process. In the diagram the large squares show the macroblock vectors to be selected and the small squares show the DCT block vectors to be selected. The three vectors are passed to a median filter which outputs the vector in the centre of the range unchanged.

A median filter is used because the same process can be performed in the decoder with no additional data transmission. The median vector is used as a prediction, and comparison with the actual vector enables a residual to be computed and coded for transmission. At the decoder the same prediction can be made and the received residual is added to re-create the original vector.

The remaining parts of Figure 5.51(b) show how the remaining three DCT block vectors are predicted from adjacent DCT block vectors. If the relevant block is only macroblock coded, that vector will be substituted.

Google Exhibit No. 1005
Page 086

Google Exhibit No. 1063
Page 0341

5.22 Shape coding

Shape coding is the process of compressing alpha or keying data. Most objects are opaque and so a binary alpha signal is adequate for the base-level shape system. For each texture pixel, an alpha bit exists forming a *binary alpha map* which is effectively a two-dimensional mask through which the object can be seen. At the decoder binary alpha data are converted to 000 or 255 levels in an eight-bit keying system.

Optionally the object can be faded in the compositing process by sending a *constant alpha value* at each VOP. As a further option, variable transparency can be supported by sending *alpha texture data*. This is coded using the usual MPEG spatial coding tools such as DCT and scanning etc.

Binary data such as alpha data do not respond to DCT coding, and another compression technique has been developed for binary alpha blocks (bab). This is known as *context-based coding* and it effectively codes the location of the boundary between alpha one bits and alpha zero bits. Clearly once the boundary is located the values of all remaining alpha bits are obvious.

The babs are raster scanned into a serial bitstream. Context coding works by attempting to predict the state of the current bit from a set of bits which have already been decoded. Figure 5.53(a) shows the set of bits, known as a context, used in an intra-coded VOP. There are ten bits in the context and so there can be 1024 different contexts. Extensive analysis of real shapes shows that for each context there is a certain probability that the current bit will be zero. This probability exists as a standardized look-up table in the encoder and decoder.

Figure 5.53(b) shows that the encoder compares the probability with a parameter called the arithmetic code value to predict the value of the alpha bit. If the prediction is correct, the arithmetic code value needs no change and nothing needs to be transmitted. However, if the predicted bit is incorrect, a change to the arithmetic code value must be transmitted, along with the position of the bit in the scan to which it applies. This change is calculated to be just big enough to make the prediction correct. In this way only prediction errors, which represent the boundary location where alpha changes, are transmitted and the remaining bits self-predict.

Figure 5.54 shows that the spatial extent of the context requires alpha bits from adjacent babs to code the present bab. This results in the concept of the *bordered bab* which incorporates a two-pixel deep border of alpha bits above and to each side. In the case of babs adjacent to top or sides of the bounding rectangle, some of the bounding bits are outside that rectangle and their value is obviously always zero.

Shape data can also be inter-coded. This is handled using a context which spreads across two VOPs. As Figure 5.55 shows, four context bits

Google Exhibit No. 1005
Page 087

Google Exhibit No. 1063
Page 0342

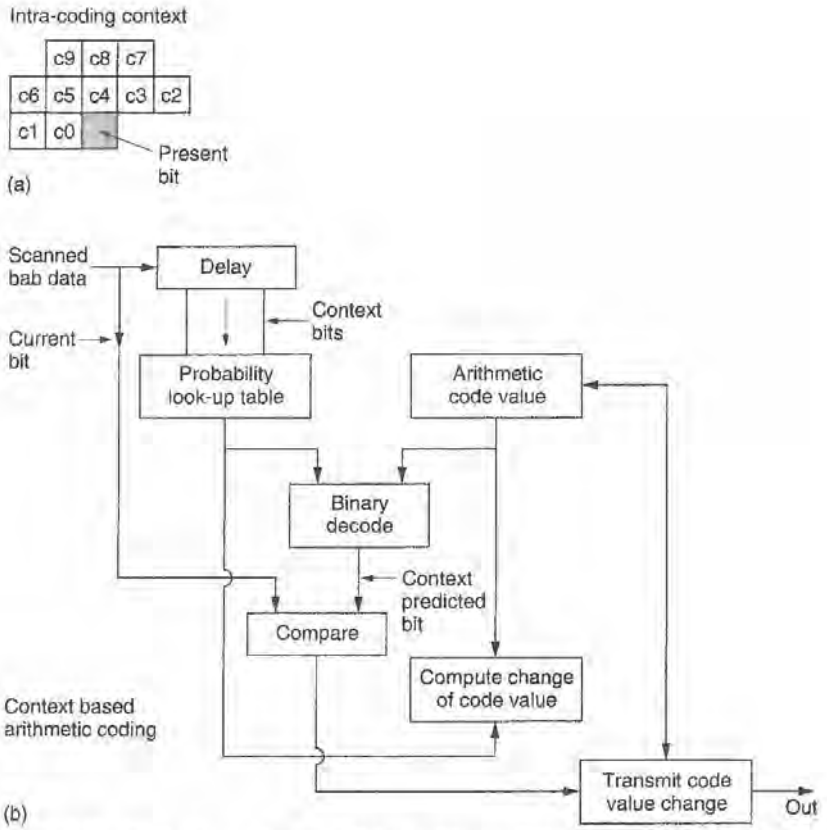


Figure 5.53 (a) The context is a set of bits used for prediction of the current bab bit. (b) The encoder compares the actual bit with the predicted value and modifies the arithmetic code in order to make the prediction correct at the decoder.

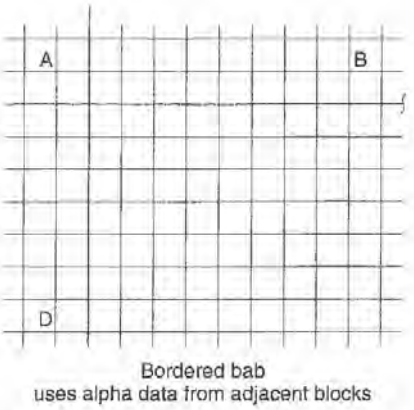


Figure 5.54 A bordered bab contains bits from adjacent babs so that context bits are available.

Google Exhibit No. 1005
Page 088

Google Exhibit No. 1063
Page 0343

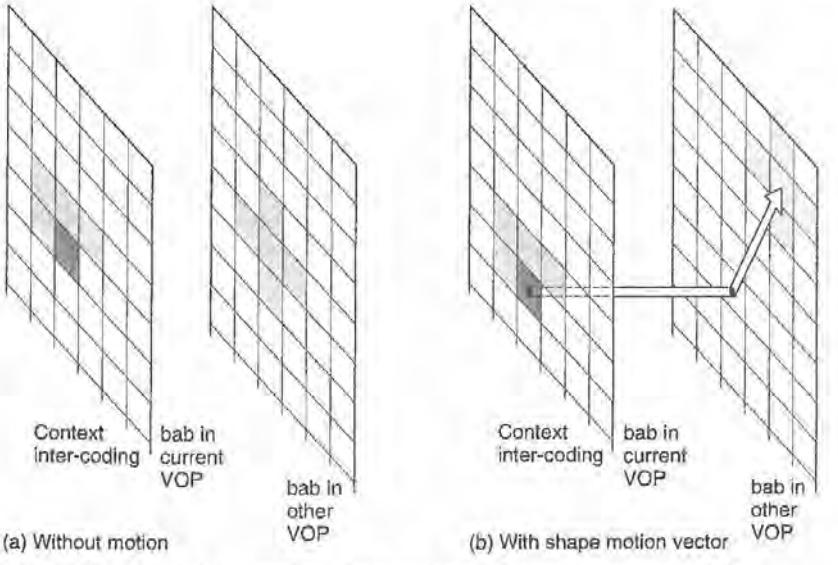


Figure 5.55 (a) In bab inter-coding, the context spreads over the current and the previous VOP. (b) Shape motion vectors shift the context with respect to the reference VOP to improve the accuracy of prediction.

are in the current VOP, whereas five context bits are in a previous VOP. This nine-bit context requires a different probability table, but otherwise coding proceeds as before. If the shape changes from one VOP to another, motion compensation can be used. A shape vector is transmitted. It will be seen from Figure 5.55(b) that this vector shifts the context relative to the previous VOP, thus altering (hopefully improving) the chances of correctly predicting the current bit.

Within a boundary block, shape vectors are highly redundant with the motion vectors of texture and texture vectors may be used to predict shape vectors.

It might be thought that binary shape data would result in objects having ratcheted edges because the key signal can only exist in pixel steps. However, texture coding can overcome this problem. If a binary keyed image is used as a predictor, the texture prediction error will deliver the necessary accuracy in the final image.

5.23 Padding

The peculiar conditions in boundary macroblocks where some of the block is inside the object and some outside requires special handling and this is the function of padding. There are two types of padding which

Google Exhibit No. 1005
Page 089

Google Exhibit No. 1063
Page 0344

perform distinct functions; texture padding and motion compensation (MC) padding.

Texture padding makes the spatial coding of boundary blocks more efficient. As the frequency coefficients of the DCT relate to the entire block area, detail outside the object can result in coefficients which are not due to the object itself. Texture padding consists of replacing those pixels in a boundary block which are outside the object with pixel values that are selected to minimize the amount of coefficient data. The exact values are unimportant as they will be discarded in the decoder.

When objects move they may also change their shape. Motion compensation will be used whereby texture and shape from another VOP is used to predict texture and shape in the current VOP. If there is a small error in the boundary data, background pixels could leak into the object causing a large prediction error. MC padding is designed to prevent this. In the case of VOPs used as anchors for inter-coding, in boundary blocks, after decoding, texture padding is replaced with MC padding. In MC padding, pixel values at the very edge of the object are copied to pixels outside the object so that small shape errors can be corrected with minor residual data.

5.24 Video object coding

Figure 5.56 shows the video object coding process. At the encoder, image analysis dissects the images into objects, in this case one small object (1) and a background object (2). The picture and shape data for object 1 are used to create a bounding rectangle whose coordinates are coded. Pixel values outside the boundary are replaced with texture padding. The texture macroblocks are then DCT coded and the shape babs are context coded. This can be on an intra- or motion-compensated inter-basis according to the type of VOP being coded. Background data are coded by another process and multiplexed into the bitstream as another object.

At the decoder the texture and shape data are decoded, and in anchor VOPs the shape data are used to strip off the texture padding and replace it with MC padding. MC padded object data are used to predict intra-coded objects. Finally the shape data key the object into the background.

Figure 5.57 shows an object coder. The object analysis of the incoming image provides texture and texture motion as well as object shape and shape motion. Texture coding takes place in much the same way as MPEG picture coding. Intra-VOPs are spatially coded, whereas P- and B-VOPs use motion-compensated prediction and a spatially coded residual is transmitted. The vectors in MPEG-4 are subject to a prediction process not found in MPEG-2. Note the use of texture padding before the

Google Exhibit No. 1005
Page 090

Google Exhibit No. 1063
Page 0345

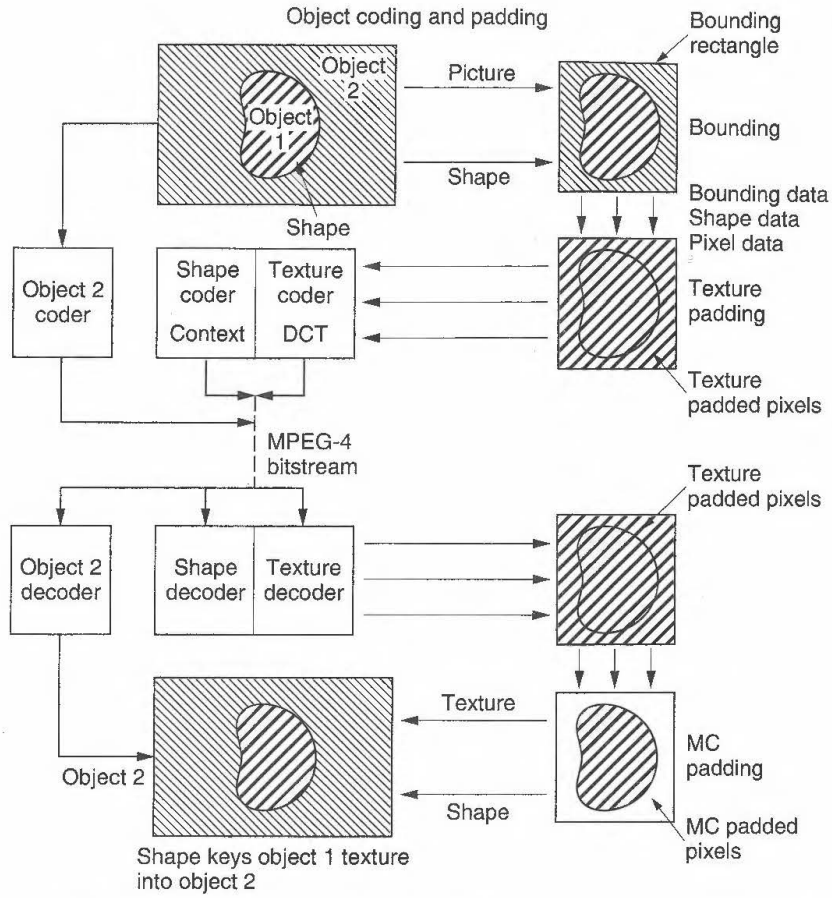


Figure 5.56 The video object coding process. Each object requires shape and texture data. One or more objects and a background are transmitted.

coder. As the decoder will use MC padding in the course of prediction, the encoder must also use it to prevent drift.

The shape data are context coded in I-VOPs and the coding will be motion compensated in P- and B-VOPs. However, the shape vectors can be predicted from other shape vectors or from the texture motion vectors where these exist. These shape vectors are coded as prediction residuals. Figure 5.58 shows how the shape vector prediction works. Around the bab to be coded are three designated shape vectors and three designated texture vectors. The six designated vectors are scanned in the order shown and the first one which is defined will be used as the predictor. Consequently if no shape vectors are defined, the texture vectors will be used automatically since these are later in the scan.

Google Exhibit No. 1005
Page 091

Google Exhibit No. 1063
Page 0346

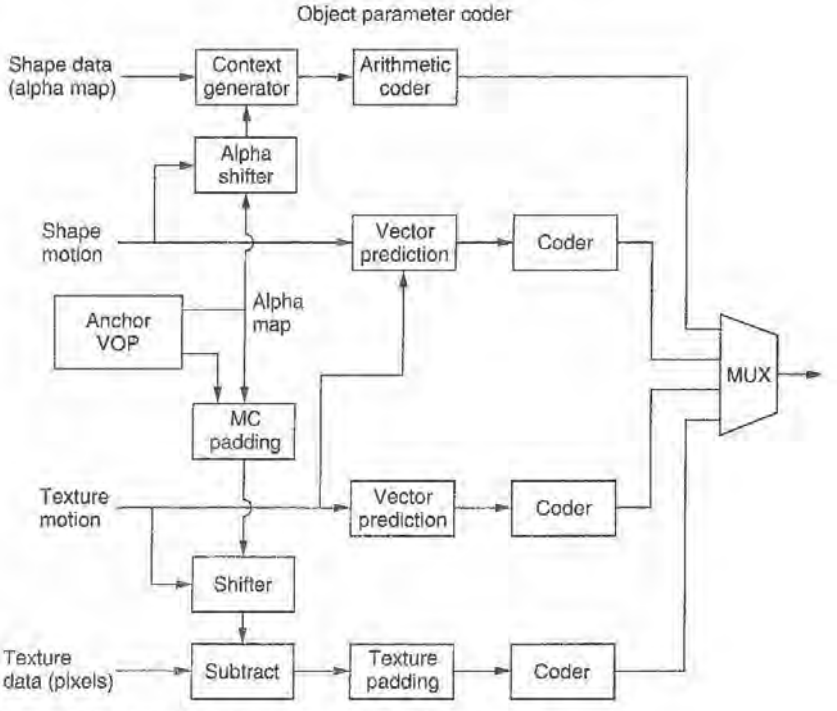


Figure 5.57 An MPEG-4 object coder. See text for details.

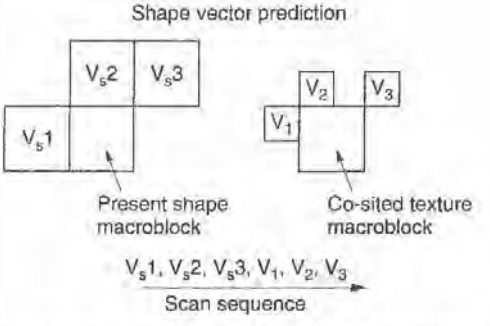


Figure 5.58 Shape vectors may be coded or taken from texture vectors. The decoder scans vectors is the sequence shown and takes the first ones available.

Shape babs can exist in a variety of forms as Figure 5.47 showed. Opaque and transparent blocks are easily and efficiently coded. Intra babs have no shape vector but need a lot of data. Inter babs may or may not contain a shape vector according to object motion and only a residual is sent. On the other hand, under certain object conditions, sufficient shape accuracy may result by sending only the shape vector.

Google Exhibit No. 1005
Page 092

Google Exhibit No. 1063
Page 0347

5.25 Two-dimensional mesh coding

Mesh coding was developed in computer graphics to aid the rendering of synthetic images with perspective. As Figure 5.59 shows, as a three-dimensional body turns, its two-dimensional appearance changes. If the body has a flat side square-on to the optical axis whose texture is described by uniformly spaced pixels, after a rotation the pixels will no longer be uniformly spaced. Nearly every pixel will have moved, and clearly an enormous amount of data would be needed to describe the motion of each one. Fortunately this is not necessary; if the geometry of the object is correctly sampled before and after the motion, then the shift of every pixel can be computed.

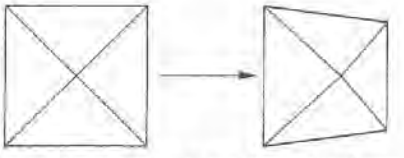


Figure 5.59 As a body turns, its shape on a two-dimensional screen may change.

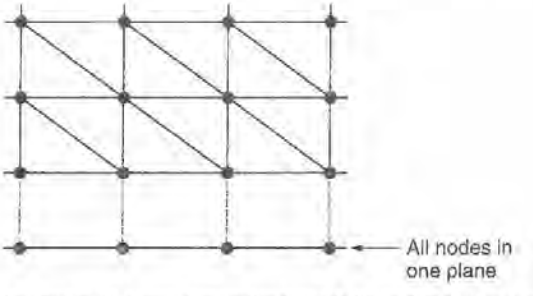


Figure 5.60 All the nodes of a two-dimensional mesh lie in the same plane.

The geometric sampling results in a structure known as a mesh. Effectively a mesh is a sampled surface. The smoother the surface, the fewer samples are necessary because the remainder of the surface can be interpolated. In MPEG-4, meshes can be two- or three-dimensional. Figure 5.60 shows a two-dimensional mesh which is a set of points known as *nodes* which must remain in the video object plane. Figure 5.61 shows a three-dimensional mesh which is a set of nodes describing a non-flat surface.

Like other MPEG-4 parameters, meshes can be scaleable. A base mesh has relatively few nodes; adding enhancement data will increase the number of nodes so that the shape is better described.

Google Exhibit No. 1005
Page 093

Google Exhibit No. 1063
Page 0348

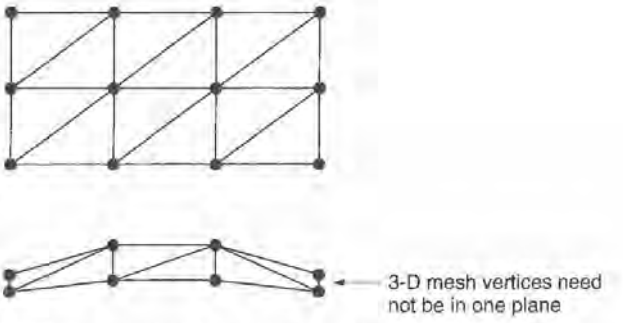


Figure 5.61 A three-dimensional mesh samples the surface of a solid body.

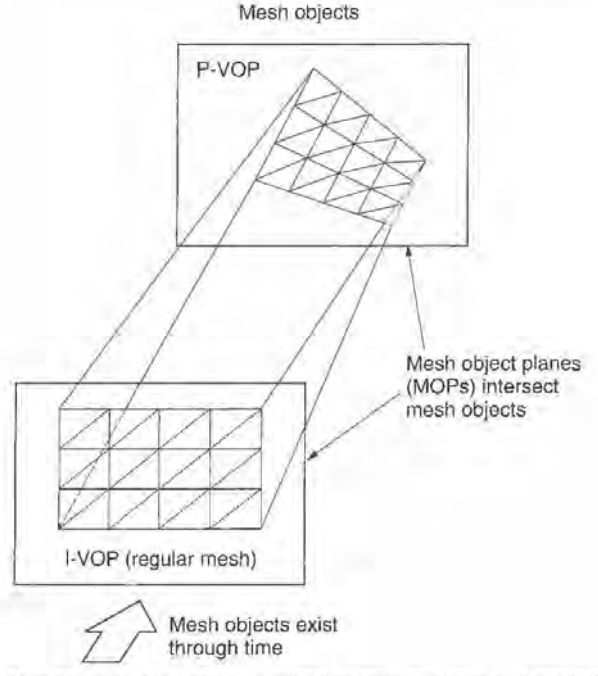


Figure 5.62 In mesh coding, an object is coded as texture, shape and mesh data. The I-VOP mesh may be regular and need few data to code it.

Figure 5.62 shows an example of how two-dimensional meshes are used to predict an object from a previous VOP. The first object may be sent as an I-VOP, including texture and alpha outline data, both of which will be intra-coded. The texture pixels in the I-VOP are uniformly spaced and so a uniform mesh can be sent as a further part of the same I-VOP. Uniform meshes require very few data. Figure 5.63(a) shows that little

Google Exhibit No. 1005
Page 094

Google Exhibit No. 1063
Page 0349

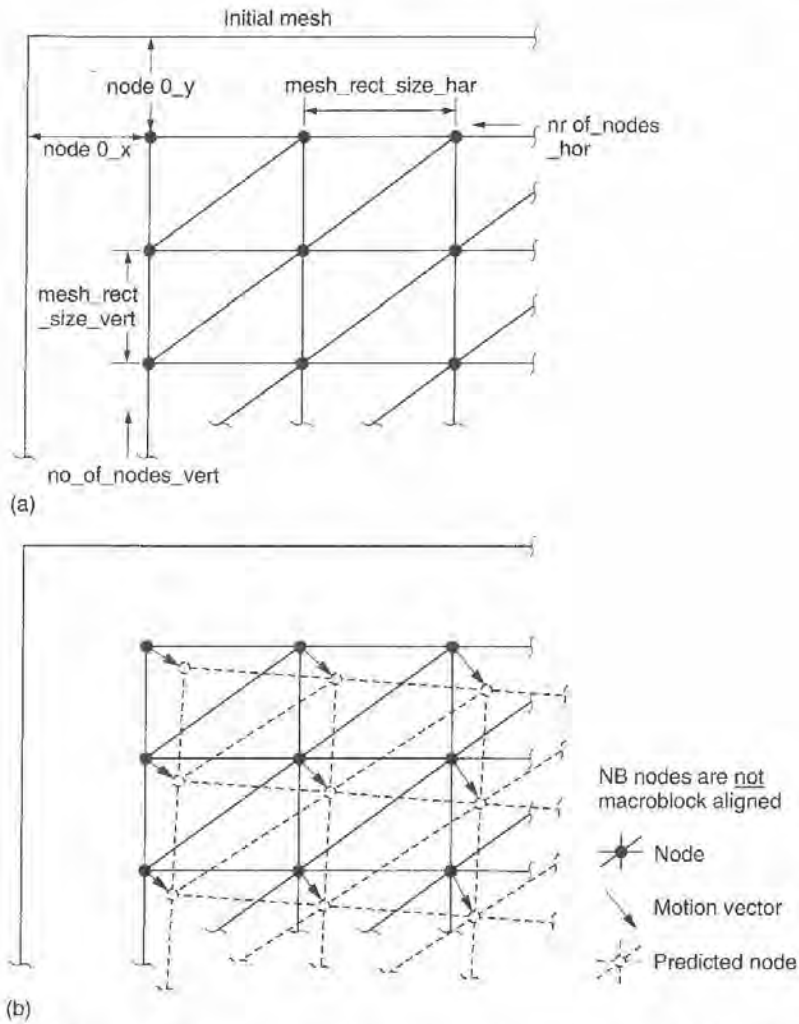


Figure 5.63 (a) The initial mesh may be regular which requires few data. (b) A mesh may be warped by transmitting vectors which displace the nodes.

more than the coordinates of the top-left corner of the mesh and the mesh pitch need be sent. There are also some choices of how the mesh boxes are divided into triangles.

As an alternative to a regular mesh, an irregular or *Delaunay* mesh may be created by specifying the location of each vertex. This will require more data initially, but may prove to be more efficient in the case of an irregular object which exists for a long time.

Figure 5.63(b) shows how the shape of the mesh can be changed by transmitting mesh vectors in P-VOPs. These specify the new position of

Google Exhibit No. 1005
Page 095

Google Exhibit No. 1063
Page 0350

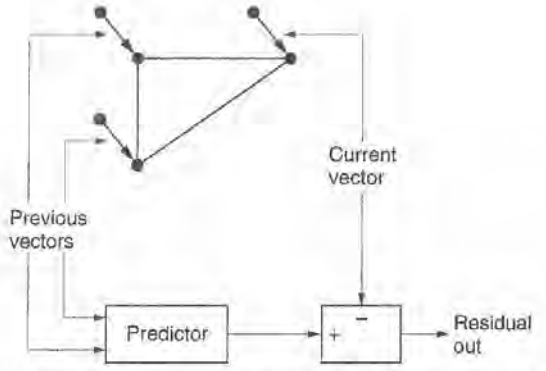


Figure 5.64 Mesh vectors are predicted from the previous two vertices of the triangle concerned.

each vertex with respect to the original position. The mesh motion vectors are predictively coded as shown in Figure 5.64. In two adjacent triangles there are two common nodes. If the first triangle has been fully coded, only one node of the second triangle remains to be coded. The vector for the remaining node is predicted by taking the average of the two common nodes and rounding to half-pixel accuracy. This prediction is subtracted from the actual vector to produce a residual which is then variable-length coded for transmission. As each node is coded, another triangle is completed and a new pair of common nodes become available for further prediction.

Figure 5.65 shows an object coding system using meshes. At (a), following the transmission of the I-VOP containing shape, outline and a regular mesh, the I-VOP and the next VOP, which will be a P-VOP, are supplied to a motion estimator. The motion estimator will determine the motion of the object from one VOP to the next. As the object moves and turns, its shape and perspective will change. The new shape may be encoded using differential alpha data which the decoder uses to change the previous shape into the current shape. The motion may also result in a perspective change. The motion estimator may recognize features in the object which are at a new location and this information is used to create mesh vectors.

At the encoder, mesh vectors are used to warp the pixels in the I-VOP to predict the new perspective. The differential shape codes are used to predict the new shape of the object. The warped, reshaped object is then compared with the actual object, and the result is a texture residual which will be DCT coded.

At the decoder, (b) the I-VOP is in memory and the mesh motion, the shape difference and the texture residual arrive as a P-VOP. The mesh motion is decoded and used to warp the object pixels taken from the

Google Exhibit No. 1005
Page 096

Google Exhibit No. 1063
Page 0351

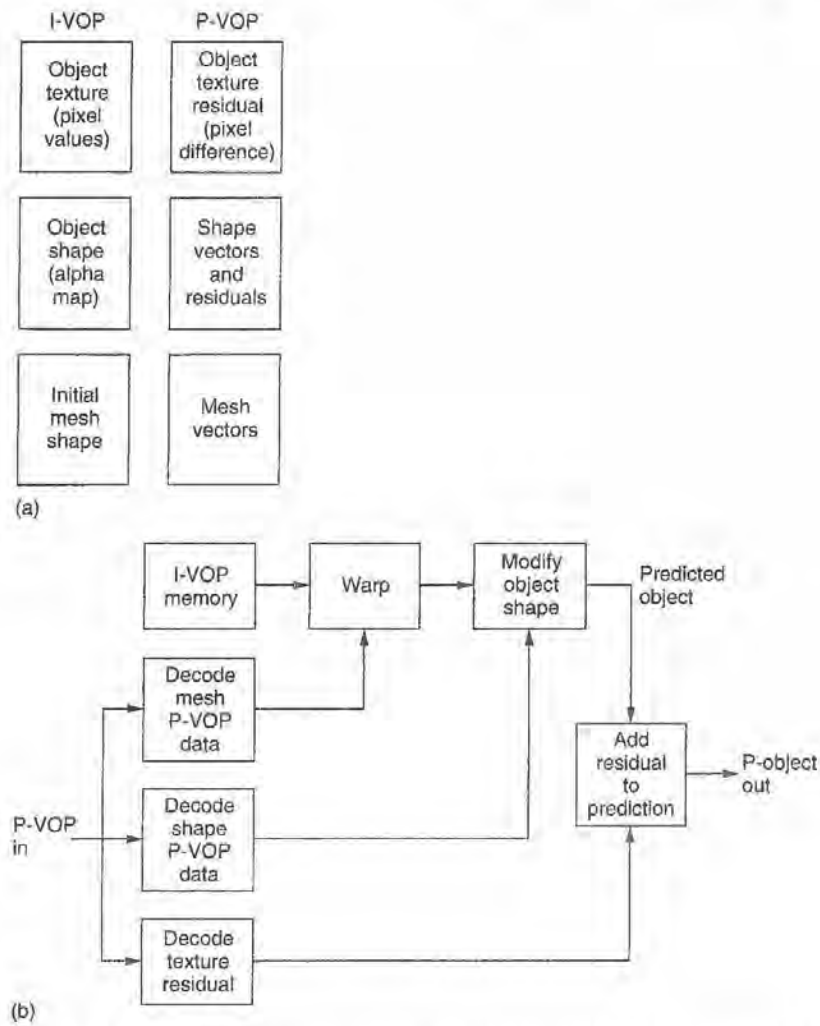


Figure 5.65 (a) Mesh coding requires texture, shape and mesh data to be multiplexed into the bitstream. (b) An object coding system using meshes. See text.

I-VOP memory. The shape difference is decoded to produce a new shape around the warped pixels, resulting in a predicted object. The texture residual is decoded and added to the predicted object to produce the final decoded object which is then composited into the output picture along with other objects.

It should be clear that by transmitting warping vectors the pixels from the I-VOP can be used to create a very accurate prediction of the texture of the P-VOP. The result is that the texture residual can be encoded with a very small number of bits.

Google Exhibit No. 1005
Page 097

Google Exhibit No. 1063
Page 0352

In both cases if the source material is video from a camera, the encoder will need sophisticated algorithms which recognize the human features and output changing animation parameters as expression and posture change.

5.30 Scaleability

MPEG-4 offers extensive support for scaleability. Given the wide range of objects which MPEG-4 can encode, offering scaleability for virtually all of them must result in considerable complexity. In general, scaleability requires the information transmitted to be decomposed into a base layer along with one or more enhancement layers. The base layer alone can be decoded to obtain a picture of a certain quality. The quality can be improved in some way by adding enhancement information.

Figure 5.86 shows a generic scaleable decoder. There is a decoder for the base layer information and a decoder for the enhancement layer. For efficient compression, the enhancement layer decoder will generally require a significant number of parameters from the base layer to assist with predictively coded data. These are provided by the *mid-processor*. Once the base and enhancement decoding is complete, the two layers are combined in a post-processor.

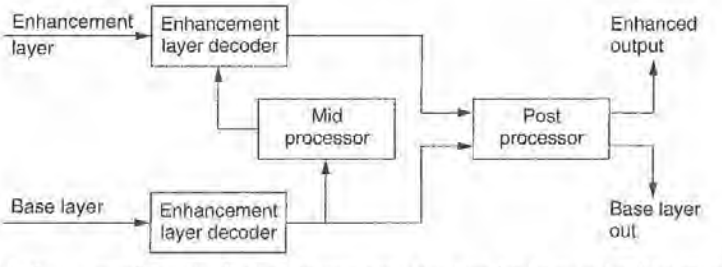


Figure 5.86 A generic scaleable decoder contains decoding for base and enhancement layers as well as a mid-processor which adapts the base layer data to the needs of the enhancement process.

Figure 5.87 shows a spatial scaleability system in which the enhancement improves resolution. Spatial scaleability may be applied to rectangular VOPs or to arbitrarily shaped objects using alpha data. In the case of irregular objects the alpha data will also need to be refined. The mid-processor will upsample the base layer images to provide base pixel data at the new, higher resolution, sampling grid. With respect to a sampling array upsampled by a factor of two, motion vectors appear half as accurate and there are one quarter as many as are needed. Consequently to improve prediction in the enhancement layer, the motion compensation must be made more accurate.

Google Exhibit No. 1005
Page 098

Google Exhibit No. 1063
Page 0353

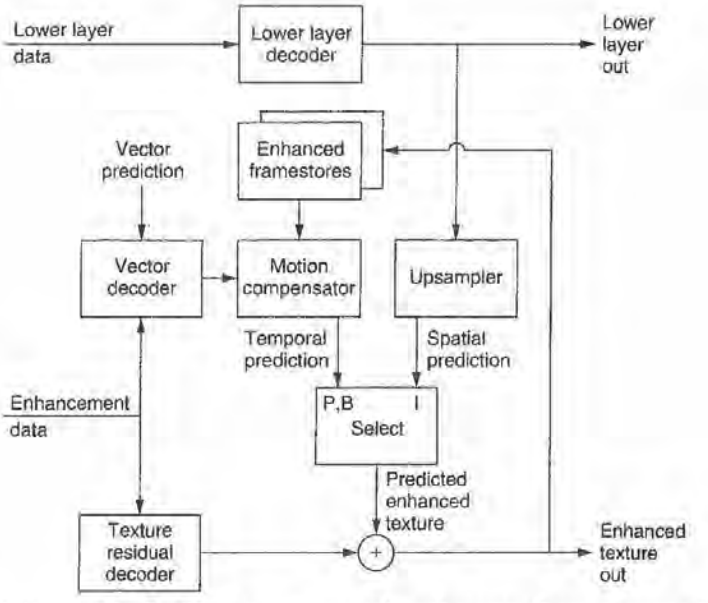


Figure 5.87 Spatial scaleability system requires the enhancement layer to improve the vector density in order to minimize the residual bit rate.

The output of the base level decoder is considered to be a spatial prediction by the enhancement process. This may be shifted by vectors from the enhancement bitstream. Figure 5.88 shows that a base level macroblock of 16 × 16 pixels predicted with one vector is upsampled to four macroblocks, each of which can be shifted with its own enhancement

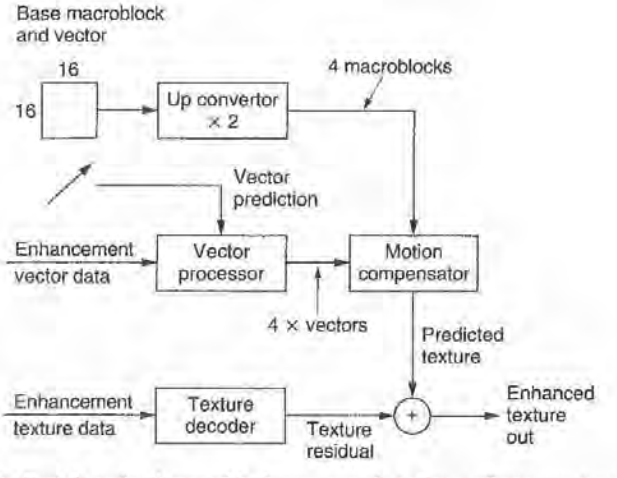


Figure 5.88 Each base level macroblock corresponds to four enhancement macroblocks. Base vectors are used to predict enhancement vectors.

Google Exhibit No. 1005
Page 099

Google Exhibit No. 1063
Page 0354

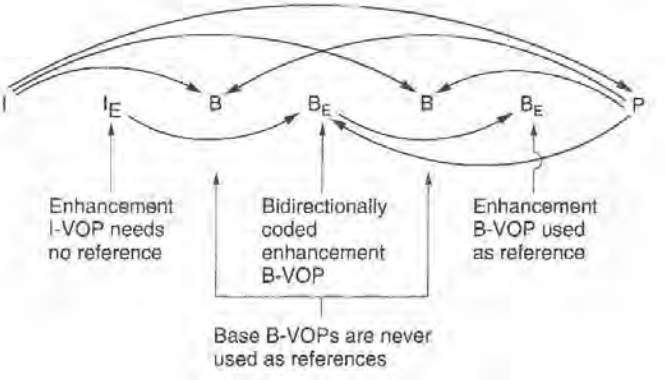


Figure 5.89 Temporal scaleability allows new VOPs to be created between base layer VOPs thereby increasing the picture rate.

vector. Upsampled base layer pixels which have been shifted by enhancement vectors form a new high-resolution prediction. The encoder transmits the spatial residual from this prediction.

Temporally scaleable decoding allows new VOPs to be created in between the VOPs of the base layer to increase the frame rate. Figure 5.89 shows that enhancement I-VOPs are intra-coded and so are decoded without reference to any other VOP just like a base I-VOP. P- and B-VOPs in the enhancement layer are motion predicted from other VOPs. These may be in the base layer or the enhancement layer according to prediction reference bits in the enhancement data. Base layer B-VOPs are never used as enhancement references, but enhancement B-VOPs can be references for other enhancement VOPs.

Figure 5.90 shows the example of an enhancement B-VOP placed between two VOPs in the base layer. Enhancement vectors are used to

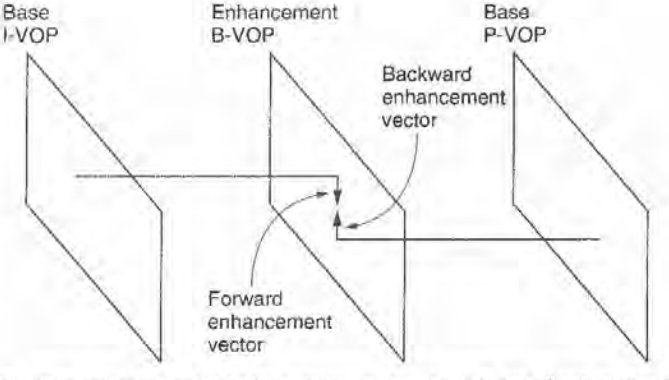


Figure 5.90 This enhancement B-VOP contains vectors to shift base level anchor VOPs. Residuals are then added to correct the prediction error.

Google Exhibit No. 1005
Page 100

Google Exhibit No. 1063
Page 0355

shift base layer pixels to new locations in the enhancement VOP. The enhancement residual is then added to these predictions to create the output VOP.

5.31 Coding artifacts

This section describes the visible results of imperfect coding. This may be where the coding algorithm is sub-optimal, where the coder latency is too short or where the compression factor in use is simply too great for the material.

Figure 5.91 shows that all MPEG coders can be simplified to the same basic model in which the predictive power of the decoder is modelled in the encoder. The current picture/object is predicted as well as possible, and the prediction error or residual is transmitted. As the decoder can make the same prediction, if the residual is correctly transmitted, the decoded output will be lossless.

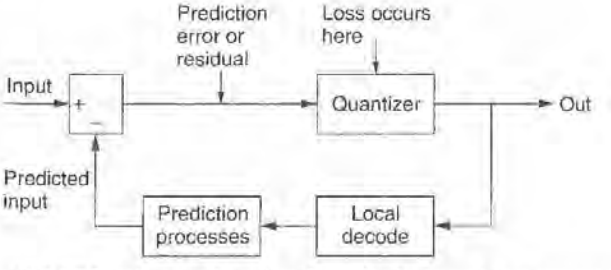


Figure 5.91 All MPEG codecs can be reduced to a lossless prediction process and a lossy coding of the prediction error or residual.

The MPEG coding family contains a large number of compression tools and many of these are lossless. Lossless coding tools include the motion compensation, zig-zag scanning and run-length coding of MPEG-1 and MPEG-2 and the context coding and various types of predictive coding used in MPEG-4. It is a simple fact that the more sophisticated the prediction, the fewer data will be needed to transmit the residual.

However, in many, perhaps most, applications of MPEG, the bit rate is not decided by the encoder, but by some other decision process which may be economically biased. All MPEG coders contain an output buffer, and if the allocated bit rate is not adequate, this will start to fill up. To prevent data loss and decoder crashes, the encoder must reduce its bit rate.

It cannot make any economy in the lossless coding. Reducing the losslessly coded information would damage the prediction mechanism

Google Exhibit No. 1005
Page 101

Google Exhibit No. 1063
Page 0356

frames. These are easy enough to code, but a huge prediction error is required to return the white frame to the next picture. The output buffer fills and heavy requantizing is employed. After a few flashes the picture has generally gone to tiles.

5.32 MPEG and concatenation

Concatenation loss occurs when the losses introduced by one codec are compounded by a second codec. All practical compressers, MPEG included, are lossy because what comes out of the decoder is not bit-identical to what went into the encoder. The bit differences are controlled so that they have minimum visibility to a human viewer.

MPEG is a toolbox which allows a variety of manipulations to be performed both in the spatial and the temporal domain. MPEG-2 has more tools than MPEG-1 and MPEG-4 has more still. There is a limit to the compression which can be used on a single picture/object, and if higher compression factors are needed, temporal coding will have to be used. The longer the run of time considered, the lower the bit rate needed, but the harder it becomes to edit.

The most editable form of MPEG is to use *I* data only. As there is no temporal coding, pure cut edits can be made between pictures. The next best thing is to use a repeating *IB* structure which is locked to the odd/even field structure. Cut edits cannot be made as the *B* pictures are bidirectionally coded and need data from both adjacent *I* pictures for decoding. The *B* picture has to be decoded prior to the edit and re-encoded after the edit. This will cause a small concatenation loss.

Beyond the *IB* structure processing gets harder. If a long GOP is used for the best compression factor, an *IBBPBBP* . . . structure results. Editing this is very difficult because the pictures are sent out of order so that bidirectional decoding can be used. MPEG allows closed GOPs where the last *B* picture is coded wholly from the previous pictures and does not need the *I* picture in the next GOP. The bitstream can be switched at this point but only if the GOP structures in the two source video signals are synchronized (makes colour framing seem easy). Consequently in practice a long GOP bitstream will need to be decoded prior to any production step. Afterwards it will need to be re-encoded.

This is known as *naive* concatenation and an enormous pitfall awaits. Unless the GOP structure of the output is identical to and synchronized with the input the results will be disappointing. The worst case is where an *I* picture is encoded from a picture which was formerly a *B* picture. It is easy enough to lock the GOP structure of a coder to a single input, but if an edit is made between two inputs, the GOP timings could well be different.

Google Exhibit No. 1005
Page 102

Google Exhibit No. 1063
Page 0357

As there are so many structures allowed in MPEG, there will be a need to convert between them. If this has to be done, it should only be in the direction which increases the GOP length and reduces the bit rate. Going the other way is inadvisable. The ideal way of converting from, say, the *IB* structure of a news system to the *IBBP* structure of an emission system is to use a recompressor. This is a kind of standards convertor which will give better results than a decode followed by an encode.

The DCT part of MPEG itself is lossless. If all the coefficients are preserved intact an inverse transform yields the same pixel data. Unfortunately this does not yield enough compression for many applications. In practice the coefficients are made less accurate by removing bits starting at the least significant end and working upwards. This process is weighted, or made progressively more aggressive as spatial frequency increases.

Small-value coefficients may be truncated to zero and large-value coefficients are most coarsely truncated at high spatial frequencies where the effect is least visible.

Figure 5.93(a) shows what happens in the ideal case where two *identical* coders are put in tandem and synchronized. The first coder quantizes the coefficients to finite accuracy and causes a loss on decoding. However, when the second coder performs the DCT calculation, the coefficients obtained will be identical to the quantized coefficients in the first coder

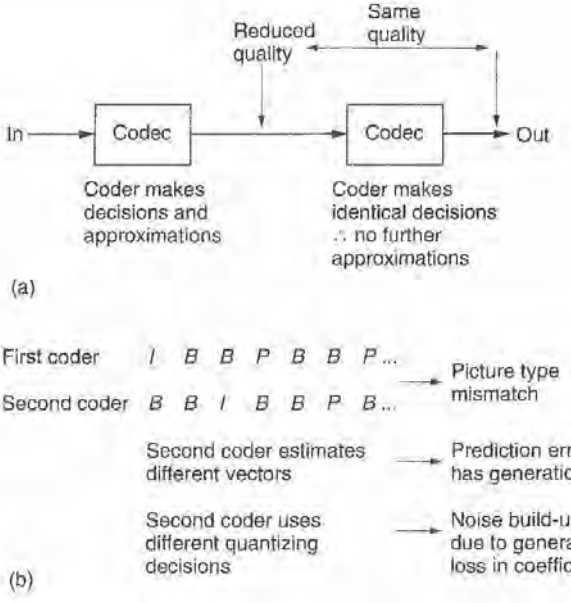


Figure 5.93 (a) Two identical coders in tandem which are synchronized make similar coding decisions and cause little loss. (b) There are various ways in which concatenated coders can produce non-ideal performance.

Google Exhibit No. 1005
Page 103

Google Exhibit No. 1063
Page 0358

and so if the second weighting and requantizing step is identical the same truncated coefficient data will result and there will be no further loss of quality.[7]

In practice this ideal situation is elusive. If the two DCTs become non-identical for any reason, the second requantizing step will introduce further error in the coefficients and the artifact level goes up. Figure 5.93(b) shows that non-identical concatenation can result from a large number of real-world effects.

An intermediate processing step such as a fade will change the pixel values and thereby the coefficients. A DVE resize or shift will move pixels from one DCT block to another. Even if there is no processing step, this effect will also occur if the two codecs disagree on where the MPEG picture boundaries are within the picture. If the boundaries are correct there will still be concatenation loss if the two codecs use different weighting.

One problem with MPEG is that the compressor design is unspecified. Whilst this has advantages, it does mean that the chances of finding identical coders is minute because each manufacturer will have his own views on the best compression algorithm. In a large system it may be worth obtaining the coders from a single supplier.

It is now increasingly accepted that concatenation of compression techniques is potentially damaging, and results are worse if the codecs are different. Clearly feeding a digital coder such as MPEG-2 with a signal which has been subject to analog compression comes into the category of worse. Using interlaced video as a source for MPEG coding is sub-optimal and using decoded composite video is even worse.

One way of avoiding concatenation is to stay in the compressed data domain. If the goal is just to move pictures from one place to another, decoding to traditional video so that an existing router can be used is not ideal, although it is substantially better than going through the analog domain.

Figure 5.94 shows some possibilities for picture transport. Clearly if the pictures exist as a compressed file on a server, a file transfer is the right way to do it as there is no possibility of loss because there has been no concatenation. File transfer is also quite indifferent to the picture format. It doesn't care about the frame rate, whether the pictures are interlaced or not or whether the colour is 4:2:0 or 4:2:2

Decoding to SDI (serial digital interface) standard is sometimes done so that existing serial digital routing can be used. This is concatenation and has to be done carefully. The compressed video can only use interlace with non-square pixels and the colour coding has to be 4:2:2 because SDI only allows that. If a compressed file has 4:2:0 the chroma has to be interpolated up to 4:2:2 for SDI transfer and then subsampled back to 4:2:0 at the second coder and this will cause generation loss. An SDI

Google Exhibit No. 1005
Page 104

Google Exhibit No. 1063
Page 0359

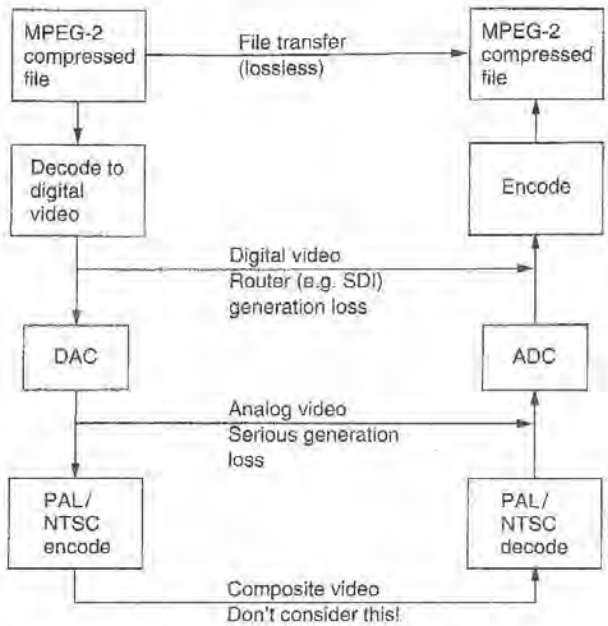


Figure 5.94 Compressed picture transport mechanisms contrasted.

transfer also can only be performed in real time, thus negating one of the advantages of compression. In short traditional SDI is not really at home with compression.

As 4:2:0 progressive scan gains popularity and video production moves steadily towards non-format-specific hardware using computers and data networks, use of the serial digital interface will eventually decline. In the short term, if an existing SDI router has to be used, one solution is to produce a bitstream which is sufficiently similar to SDI that a router will pass it. One example of this is known as SDTI. The signal level, frequency and impedance of SDTI is pure SDI, but the data protocol is different so that a bit accurate file transfer can be performed. This has two advantages over SDI. First, the compressed data format can be anything appropriate and non-interlaced and/or 4:2:0 can be handled in any picture size, aspect ratio or frame rate. Second, a faster than real-time transfer can be used depending on the compression factor of the file.

Equipment which allows this is becoming available and its use can mean that the full economic life of a SDI routing installation can be obtained.

An improved way of reducing concatenation loss has emerged from the ATLANTIC research project.[8] Figure 5.95 shows that the second encoder in a concatenated scheme does not make its own decisions from the incoming video, but is instead steered by information from the first

Google Exhibit No. 1005
Page 105

Google Exhibit No. 1063
Page 0360

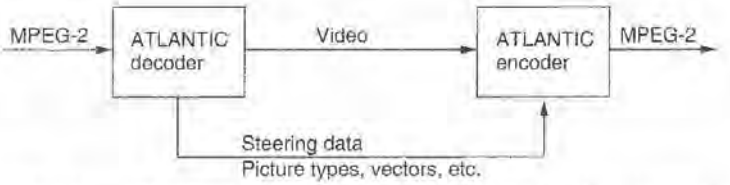


Figure 5.95 In an ATLANTIC system, the second encoder is steered by information from the decoder.

bitstream. As the second encoder has less intelligence, it is known as a *dim* encoder.

The information bus carries all the structure of the original MPEG-2 bitstream which would be lost in a conventional decoder. The ATLANTIC decoder does more than decode the pictures. It also places on the information bus all parameters needed to make the dim encoder re-enact what the initial MPEG-2 encode did as closely as possible.

The GOP structure is passed on so that pictures are re-encoded as the same type. Positions of macroblock boundaries become identical so that DCT blocks contain the same pixels and motion vectors relate to the same screen data. The weighting and quantizing tables are passed so that coefficient truncation is identical. Motion vectors from the original bitstream are passed on so that the dim encoder does not need to perform motion estimation. In this way predicted pictures will be identical to the original prediction and the prediction error data will be the same.

One application of this approach is in recompression, where an MPEG-2 bitstream has to have its bit rate reduced. This has to be done by heavier requantizing of coefficients, but if as many other parameters as possible can be kept the same, such as motion vectors, the degradation will be minimized. In a simple recompressor just requantizing the coefficients means that the predictive coding will be impaired. In a proper encode, the quantizing error due to coding say an *I* picture is removed from the *P* picture by the prediction process. The prediction error of *P* is obtained by subtracting the decoded *I* picture rather than the original *I* picture.

In simple recompression this does not happen and there may be a tolerance buildup known as drift.[9] A more sophisticated recompressor will need to repeat the prediction process using the decoded output pictures as the prediction reference.

MPEG-2 bitstreams will often be decoded for the purpose of switching. Local insertion of commercial breaks into a centrally originated bitstream is one obvious requirement. If the decoded video signal is switched, the information bus must also be switched. At the switch point identical re-encoding becomes impossible because prior pictures required for predictive coding will have disappeared. At this point the dim encoder

Google Exhibit No. 1005
Page 106

Google Exhibit No. 1063
Page 0361

has to become bright again because it has to create an MPEG-2 bitstream without assistance.

It is possible to encode the information bus into a form which allows it to be invisibly carried in the serial digital interface. Where a production process such as a vision mixer or DVE performs no manipulation, i.e. becomes bit transparent, the subsequent encoder can extract the information bus and operate in 'dim' mode. Where a manipulation is performed, the information bus signal will be corrupted and the encoder has to work in 'bright' mode. The encoded information signal is known as a 'mole'[10] because it burrows through the processing equipment!

There will be a generation loss at the switch point because the re-encode will be making different decisions in bright mode. This may be

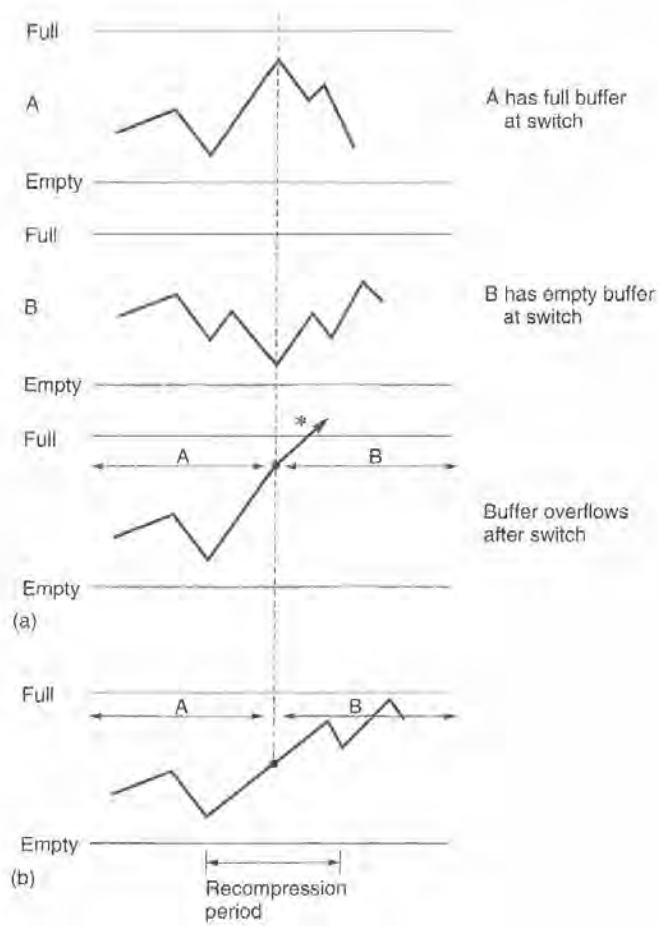


Figure 5.96 (a) A bitstream switch at a different level of buffer occupancy can cause a decoder to overflow. (b) Recompression after a switch to return to correct buffer occupancy.

Google Exhibit No. 1005
Page 107

Google Exhibit No. 1063
Page 0362

difficult to detect because the human visual system is slow to react to a vision cut and defects in the first few pictures after a cut are masked.

In addition to the video computation required to perform a cut, the process has to consider the buffer occupancy of the decoder. A downstream decoder has finite buffer memory, and individual encoders model the decoder buffer occupancy to ensure that it neither overflows nor underflows. At any instant the decoder buffer can be nearly full or nearly empty without a problem provided there is a subsequent correction. An encoder which is approaching a complex *I* picture may run down the buffer so it can send a lot of data to describe that picture. Figure 5.96(a) shows that if a decoder with a nearly full buffer is suddenly switched to an encoder which has been running down its buffer occupancy, the decoder buffer will overflow when the second encoder sends a lot of data.

An MPEG-2 switcher will need to monitor the buffer occupancy of its own output to avoid overflow of downstream decoders. Where this is a possibility the second encoder will have to recompress to reduce the output bit rate temporarily. In practice there will be a recovery period where the buffer occupancy of the newly selected signal is matched to that of the previous signal. This is shown in Figure 5.96(b).

References

1. Kelly, D.H., Visual processing of moving stimuli. *J. Opt. Soc. America*, **2** 216–225 (1985)
2. Uyttendaele, A., Observations on scanning formats. Presented at HDTV '97 Montreux (June 1997)
3. Shapiro, J.M., Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. SP*, **41** 3445–3462 (1993)
4. Taubin, G. and Rossignac, J., Geometric compression through topological surgery. *ACM Trans. on Graphics*, 88–115 (April 1998)
5. de Boor, C., *A Practical Guide to Splines*, Berlin: Springer (1978)
6. Taubin, G., Guezic, A., Horn, W.P. and Lazarus, F., *Progressive forest split compression, Proc. Siggraph '98*, 123–132 (1998)
7. Stone, J. and Wilkinson, J., Concatenation of video compression systems. Presented at 137th SMPTE Tech. Conf., New Orleans (1995)
8. Wells, N.D., The ATLANTIC project: Models for programme production and distribution. *Proc. Euro. Conf. Multimedia Applications Services and Techniques (ECMAST)*, 243–253 (1996)
9. Werner, O., Drift analysis and drift reduction for multiresolution hybrid video coding. *Image Communication*, **8**, 387–409 (1996)
10. Knee, M.J. and Wells, N.D., Seamless concatenation – a 21st century dream. Presented at Int. Television. Symp., Montreux (1997)

Google Exhibit No. 1005
Page 108

Google Exhibit No. 1063
Page 0363

6

Program and transport streams

After compression, audio and video signals are simply data and only differ from generic data in that they relate to real-time signals. MPEG bitstreams are basically means of transporting data whilst allowing the time axis of the original signal to be re-created.

6.1 Introduction

There are two basic applications of MPEG bitstreams – recording and transmission – and these have quite different requirements. In the multichannel recording application the encoders and decoders can all share the same clock. The transport speed can be altered so that the incoming bit rate can be slaved to a reference generated by the decoder itself.

In the case of multichannel transmission the program sources are not necessarily synchronous. The source of timing is the encoder and the decoder must synchronize or genlock to that. Thus the difference between a program stream and a transport stream is that the latter must contain additional synchronizing information to lock the encoder and decoder clocks together independently in each program.

6.2 Packets and time stamps

The video elementary stream is an endless bitstream representing pictures which are not necessarily in the correct order and which take a variable length of time to transmit. Storage and transmission systems prefer discrete blocks of data and so elementary streams are packetized to form a PES (packetized elementary stream). Audio elementary streams

Google Exhibit No. 1005
Page 109

Google Exhibit No. 1063
Page 0364

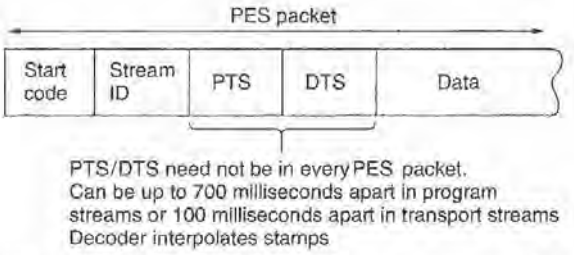


Figure 6.1 A PES packet structure is used to break up the continuous elementary stream.

are also packetized. A packet is shown in Figure 6.1. It begins with a header containing an unique packet start code and a code which identifies the type of data stream. Optionally the packet header also may contain one or more time stamps which are used for synchronizing the video decoder to real time and for obtaining lip-sync.

Figure 6.2 shows that a time stamp is a sample of the state of a counter which is driven by a 90 kHz clock. This is obtained by dividing down the master 27 MHz clock of MPEG-2. There are two types of time stamp: PTS and DTS. These are abbreviations for presentation time stamp and decode time stamp. A presentation time stamp determines when the associated picture should be displayed on the screen, whereas a decode time stamp determines when it should be decoded. In bidirectional coding these times can be quite different.

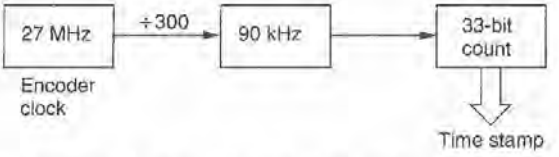


Figure 6.2 Time stamps are the result of sampling a counter driven by the encoder clock.

Audio packets only have presentation time stamps. Clearly if lip-sync is to be obtained, the audio and the video streams of a given program must have been locked to the same master 27 MHz clock and the time stamps must have come from the same counter driven by that clock.

With reference to Figure 6.3, the GOP begins with an *I* picture, and then *P1* is sent out of sequence prior to the *B* pictures. *P1* has to be decoded before *B1* and *B2* can be decoded. As only one picture can be decoded at a time, the *I* picture is decoded at time *N*, but not displayed until time *N*

Google Exhibit No. 1005
Page 110

Google Exhibit No. 1063
Page 0365

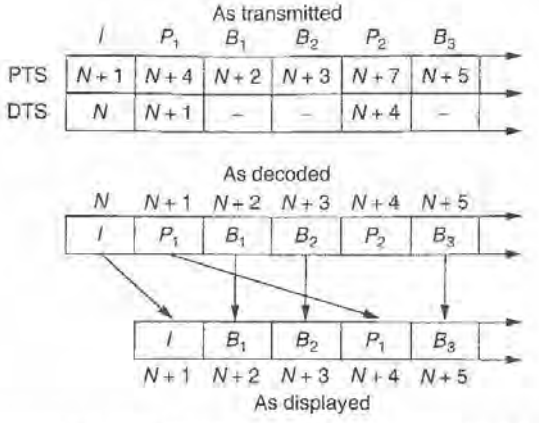


Figure 6.3 An example of using PTS/DTS to synchronize bidirectional decoding.

+ 1. As the *I* picture is being displayed, *P1* is being decoded at $N + 1$. *P1* will be stored in RAM. At time $N + 2$, *B1* is decoded and displayed immediately. For this reason *B* pictures need only PTS. At $N + 3$, *B2* is decoded and displayed. At $N + 4$, *P1* is displayed, hence the large difference between PTS and DTS in *P1*. Simultaneously *P2* is decoded and stored ready for the decoding of *B3* and so on.

In practice the time between input pictures is constant and so there is a certain amount of redundancy in the time stamps. Consequently PTS/DTS need not appear in every PES packet. Time stamps can be up to 700 ms apart in program streams and up to 100 ms apart in transport streams. As each picture type (*I*, *P* or *B*) is flagged in the bitstream, the decoder can infer the PTS/DTS for every picture from the ones actually transmitted.

Figure 6.4 shows that one or more PES packets can be assembled in a pack whose header contains a sync pattern and a system clock reference code which allows the decoder to re-create the encoder clock. Clock reference operation is described in section 6.4.

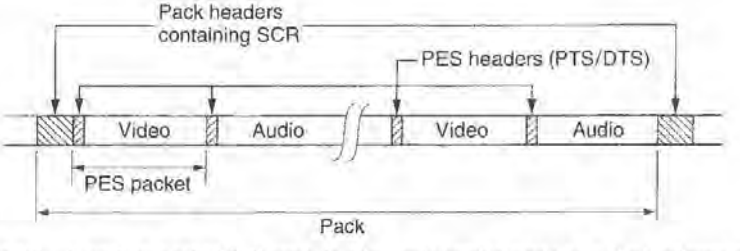


Figure 6.4 A pack is a set of PES packets. The pack header contains a clock reference code.

Google Exhibit No. 1005
Page 111

Google Exhibit No. 1063
Page 0366

6.3 Transport streams

The MPEG-2 transport stream is intended to be a multiplex of many TV programs with their associated sound and data channels, although a single program transport stream (SPTS) is possible. The transport stream is based upon packets of constant size so that adding error-correction codes and interleaving[1] in a higher layer is eased. Figure 6.5 shows that these are always 188 bytes long. Transport stream packets should not be confused with PES packets which are larger and vary in size.

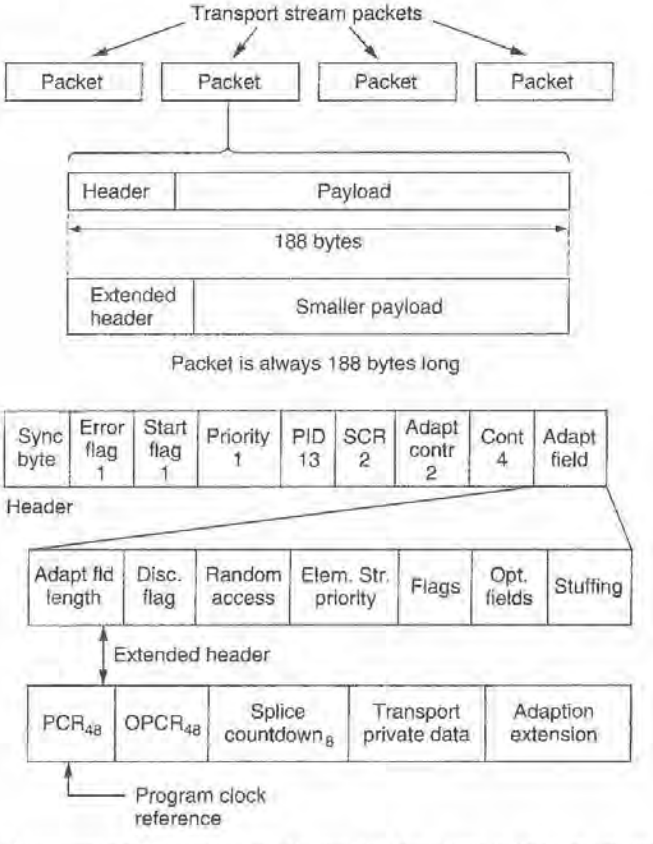


Figure 6.5 Transport stream packets are always 188 bytes long to facilitate multiplexing and error correction.

Transport stream packets always begin with a header. The remainder of the packet carries data known as the payload. For efficiency, the normal header is relatively small, but for special purposes the header may be extended. In this case the payload gets smaller so that the overall size of the packet is unchanged.

Google Exhibit No. 1005
Page 112

Google Exhibit No. 1063
Page 0367

The header begins with a sync byte which is an unique pattern detected by a demultiplexer. A transport stream may contain many different elementary streams and these are identified by giving each an unique 13-bit Packet Identification Code or PID which is included in the header. A multiplexer seeking a particular elementary stream simply checks the PID of every packet and accepts those which match, rejecting the rest.

In a multiplex there may be many packets from other programs in between packets of a given PID. To help the demultiplexer, the packet header contains a continuity count. This is a four-bit value which increments at each new packet having a given PID.

This approach allows statistical multiplexing as it does not matter how many or how few packets have a given PID; the demux will still find them. Statistical multiplexing has the problem that it is virtually impossible to make the sum of the input bit rates constant. Instead the multiplexer aims to make the average data bit rate slightly less than the maximum and the overall bit rate is kept constant by adding stuffing or null packets. These packets have no meaning, but simply keep the bit rate constant. Null packets always have a PID of 8191 (all ones) and the demultiplexer discards them.

6.4 Clock references

A transport stream is a multiplex of several TV programs and these may have originated from widely different locations. It is impractical to expect all the programs in a transport stream to be synchronous and so the stream is designed from the outset to allow asynchronous programs. A decoder running from a transport stream has to genlock to the encoder and the transport stream has to have a mechanism to allow this to be done independently for each program. The synchronizing mechanism is called Program Clock Reference (PCR).

In program streams all the programs must be synchronous so that only one clock is required at the decoder. In this case the synchronizing mechanism is called System Clock Reference (SCR).

Figure 6.6 shows how the PCR/SCR system works. The goal is to re-create at the decoder a 27 MHz clock which is synchronous with that at the encoder. The encoder clock drives a forty-eight-bit counter which continuously counts up to the maximum value before overflowing and beginning again.

A transport stream multiplexer will periodically sample the counter and place the state of the count in an extended packet header as a PCR (see Figure 6.5). The demultiplexer selects only the PIDs of the required program, and it will extract the PCRs from the packets in which they were inserted. In a program stream the count is placed in a pack header as an SCR which the decoder can identify.

Google Exhibit No. 1005
Page 113

Google Exhibit No. 1063
Page 0368

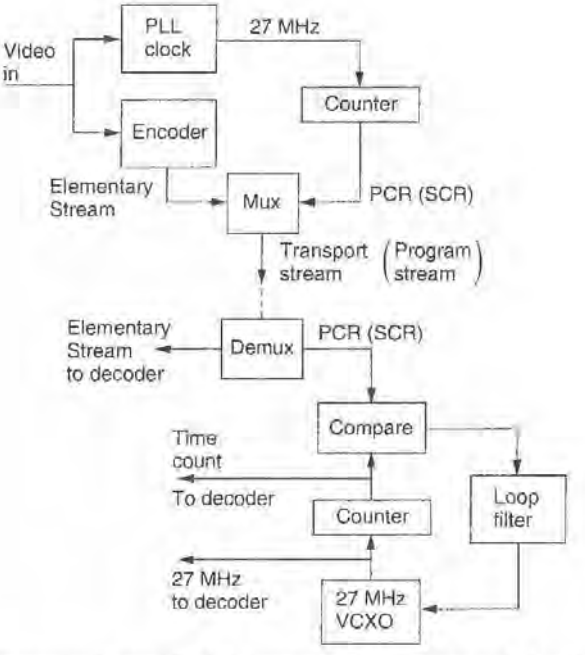


Figure 6.6 Program or System Clock Reference codes regenerate a clock at the decoder. See text for details.

The PCR/SCR codes are used to control a numerically locked loop (NLL). This is similar to a phase-locked loop, except that the two phases concerned are represented by the state of a binary number. The NLL contains a 27 MHz VCXO (voltage controlled crystal oscillator), a variable-frequency oscillator based on a crystal which has a relatively small frequency range.

The VCXO drives a forty-eight-bit counter in the same way as in the encoder. The state of the counter is compared with the contents of the PCR/SCR and the difference is used to modify the VCXO frequency. When the loop reaches lock, the decoder counter would arrive at the same value as is contained in the PCR/SCR and no change in the VCXO would then occur. In practice the transport stream packets will suffer from transmission jitter and this will create phase noise in the loop. This is removed by the loop filter so that the VCXO effectively averages a large number of phase errors.

A heavily damped loop will reject jitter well, but will take a long time to lock. Lock-up time can be reduced when switching to a new program if the decoder counter is jammed to the value of the first PCR received in the new program. The loop filter may also have its time constants shortened during lock-up. Once a synchronous 27 MHz clock is available

Google Exhibit No. 1005
Page 114

Google Exhibit No. 1063
Page 0369

at the decoder, this can be divided down to provide the 90 kHz clock which drives the time stamp mechanism. The entire timebase stability of the decoder is no better than the stability of the clock derived from PCR/SCR. MPEG-2 sets standards for the maximum amount of jitter which can be present in PCRs in a real transport stream.

6.5 Program Specific Information (PSI)

In a real transport stream, each elementary stream has a different PID, but the demultiplexer has to be told what these PIDs are and what audio belongs with what video before it can operate. This is the function of PSI. Figure 6.7 shows the structure of PSI. When a decoder powers up, it knows nothing about the incoming transport stream except that it must search for all packets with a PID of zero. PID zero is reserved for the Program Association Table (PAT). The PAT is transmitted at regular intervals and contains a list of all the programs in this transport stream. Each program is further described by its own Program Map Table (PMT) and the PIDs of of the PMTs are contained in the PAT.

Figure 6.7 also shows that the PMTs fully describe each program. The PID of the video elementary stream is defined, along with the PID(s) of the associated audio and data streams. Consequently when the viewer selects a particular program, the demultiplexer looks up the program

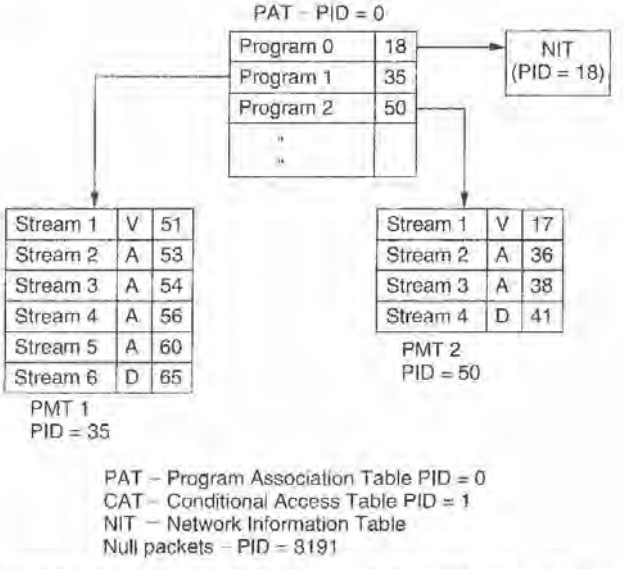


Figure 6.7 MPEG-2 Program Specific Information (PSI) is used to tell a demultiplexer what the transport stream contains.

Google Exhibit No. 1005
Page 115

Google Exhibit No. 1063
Page 0370

number in the PAT, finds the right PMT and reads the audio, video and data PIDs. It then selects elementary streams having these PIDs from the transport stream and routes them to the decoders.

Program 0 of the PAT contains the PID of the Network Information Table (NIT). This contains information about what other transport streams are available. For example, in the case of a satellite broadcast, the NIT would detail the orbital position, the polarization, carrier frquency and modulation scheme. Using the NIT a set-top box could automatically switch between transport streams.

Apart from 0 and 8191, a PID of 1 is also reserved for the Conditional Access Table (CAT). This is part of the access control mechanism needed to support pay per view or subscription viewing.

6.6 Multiplexing

A transport stream multiplexer is a complex device because of the number of functions it must perform. A fixed multiplexer will be considered first. In a fixed multiplexer, the bit rate of each of the programs must be specified so that the sum does not exceed the payload bit rate of the transport stream. The payload bit rate is the overall bit rate less the packet headers and PSI rate.

In practice the programs will not be synchronous to one another, but the transport stream must produce a constant packet rate given by the bit rate divided by 188 bytes, the packet length. Figure 6.8 shows how this is handled. Each elementary stream entering the multiplexer passes through a buffer which is divided into payload-sized areas. Note that

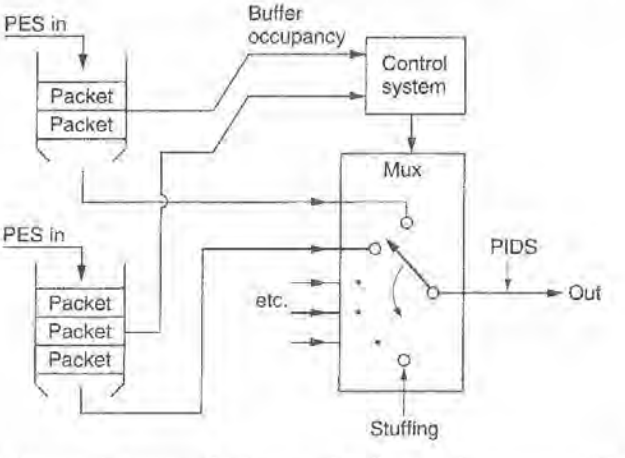


Figure 6.8 A transport stream multiplexer can handle several programs which are asynchronous to one another and to the transport stream clock. See text for details.

Google Exhibit No. 1005
Page 116

Google Exhibit No. 1063
Page 0371

periodically the payload area is made smaller because of the requirement to insert PCR.

MPEG-2 decoders also have a quantity of buffer memory. The challenge to the multiplexer is to take packets from each program in such a way that neither its own buffers nor the buffers in any decoder either overflow or underflow. This requirement is met by sending packets from all programs as evenly as possible rather than bunching together a lot of packets from one program. When the bit rates of the programs are different, the only way this can be handled is to use the buffer contents indicators. The fuller a buffer is, the more likely it should be that a packet will be read from it. This a buffer content arbitrator can decide which program should have a packet allocated next.

If the sum of the input bit rates is correct, the buffers should all slowly empty because the overall input bit rate has to be less than the payload bit rate. This allows for the insertion of Program Specific Information. While PATs and PMTs are being transmitted, the program buffers will fill up again. The multiplexer can also fill the buffers by sending more PCRs as this reduces the payload of each packet. In the event that the multiplexer has sent enough of everything but still cannot fill a packet then it will send a null packet with a PID of 8191. Decoders will discard null packets and as they convey no useful data, the multiplexer buffers will all fill while null packets are being transmitted.

In a statistical multiplexer or statmux, the bit rate allocated to each program can vary dynamically. Figure 6.9 shows that there must be tight connection between the statmux and the associated compressors. Each compressor has a buffer memory which is emptied by a demand clock from the statmux. In a normal, fixed bit rate, coder the buffer contents feeds back and controls the requantizer. In statmuxing this process is less severe and only takes place if the buffer is very close to full, because the buffer content is also fed to the statmux.

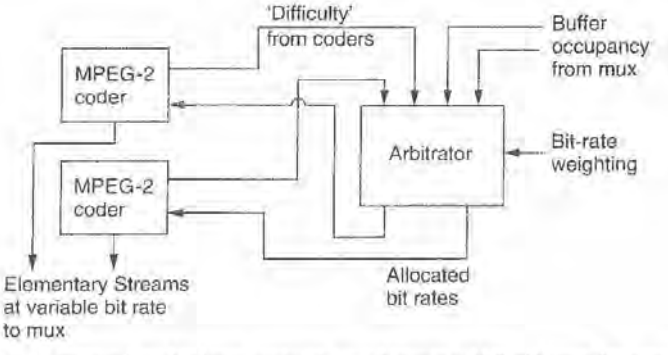


Figure 6.9 A statistical multiplexer contains an arbitrator which allocates bit rate to each program as a function of program difficulty.

Google Exhibit No. 1005
Page 117

Google Exhibit No. 1063
Page 0372

The statmux contains an arbitrator which allocates packets to the program with the fullest buffer. Thus if a particular program encounters difficult material it will produce large prediction errors and begin to fill its output buffer. This will cause the statmux to allocate more packets to that program. In order to fill more packets, the statmux clocks more data out of that buffer, causing the level to fall again. Of course, this is only possible if the other programs in the transport stream are handling typical video.

In the event that several programs encounter difficult material at once, clearly the buffer contents will rise and the requantizing mechanism will have to operate.

6.7 Remultiplexing

In real life a program creator may produce a transport stream which carries all its programs simultaneously. A service provider may take in several such streams and create its own transport stream by selecting different programs from different sources. In an MPEG-2 environment this requires a remultiplexer, also known as a transmultiplexer. Figure 6.10 shows what a remultiplexer does.

Remultiplexing is easier when all the incoming programs have the same bit rate. If a suitable combination of programs is selected it is obvious that the output transport stream will always have sufficient bit rate. Where statistical multiplexing has been used, there is a possibility that the sum of the bit rates of the selected programs will exceed the bit rate of the output transport stream. To avoid this, the remultiplexer will have to employ recompression.

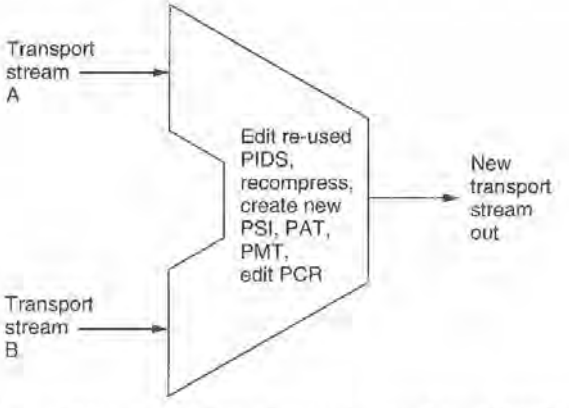


Figure 6.10 A remultiplexer creates a new transport stream from selected programs in other transport streams.

Google Exhibit No. 1005
Page 118

Google Exhibit No. 1063
Page 0373

Recompression requires a partial decode of the bitstream to identify the DCT coefficients. These will then be requantized to reduce the bit rate until it is low enough to fit the output transport stream.

Remultiplexers have to edit the Program Specific Information (PSI) such that the Program Association Table (PAT) and the Program Map Tables (PMT) correctly reflect the new transport stream content. It may also be necessary to change the packet identification codes (PIDs) since the incoming transport streams could have inadvertently used the same values.

When Program Clock Reference (PCR) data are included in an extended packet header, they represent a real-time clock count and if the associated packet is moved in time the PCR value will be wrong. Remultiplexers have to recreate a new multiplex from a number of other multiplexes and it is inevitable that this process will result in packets being placed in different locations in the output transport stream than they had in the input. In this case the remultiplexer must edit the PCR values so that they reflect the value the clock counter would have had at the location the packet at which the packet now resides.

Reference

1. Watkinson, J.R., *The Art of Digital Video*, second edition, Ch. 6, Oxford: Focal Press (1994)

Google Exhibit No. 1005
Page 119

Google Exhibit No. 1063
Page 0374

7

MPEG applications

7.1 Introduction

It should be borne in mind that however exciting it may be, MPEG is only a technology. Technology *per se* is not useful and only becomes useful when it is incorporated into an appropriate and affordable product or service. If the use is not appropriate or affordable, there is no point in blaming the technology.

In this chapter a number of useful applications of MPEG are outlined. In each case designers have followed the rules of high technology to design systems which maximize the potential of the technology whilst protecting against its weaknesses.

MPEG is an information technology. It cannot deliver anything physical: you get the movie but no popcorn. The strength of MPEG is that it delivers audio-visual material with economy of bandwidth. Its weaknesses are sensitivity to bit errors and the requirement for near-real-time transmission. The variable-length coding of MPEG can lose synchronization in the presence of a single bit error. Decoders can take a considerable time to recover from a buffer underflow or overflow.

Wherever bandwidth or storage capacity is in short supply, MPEG is useful. Where bandwidth is plentiful, as in optical fibre networks, MPEG is pointless. As the capacity and economics of storage devices continue to increase, MPEG becomes less important.

MPEG will remain important in any area where there will be a permanent shortage of bandwidth; this includes any mobile equipment or service which must use radio communication.

Figure 7.1 shows that well-engineered applications of MPEG will always include an error-correction system appropriate for the characteristics of the channel and suitable mechanisms to ensure that decoder

Google Exhibit No. 1005
Page 120

Google Exhibit No. 1063
Page 0375

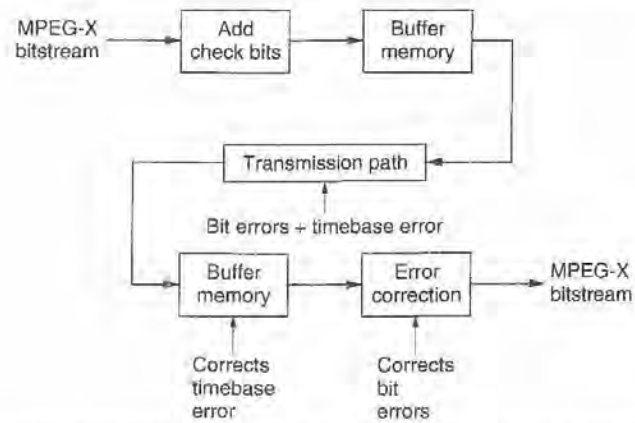


Figure 7.1 Any real-time MPEG delivery system must include means to maintain a low error rate and buffering to deliver data within the decoder's time window.

buffers neither overflow nor underflow. Constant-bit rate channels having constant delay are easy to implement because the buffering mechanisms of the standard MPEG encoders and decoders are able to operate with no further assistance. Digital TV transmitters provide essentially constant bit-rate because this is related to the standardized (and regulated) channel bandwidth allowed.

In packet networks the delivery delay is not constant because other types of messages are multiplexed with the audio-visual message of interest. Variable delay channels cause MPEG decoders difficulty. Not only is there a greater danger of buffer overflow or underflow, but the program clock reference information is no longer correct if a packet is

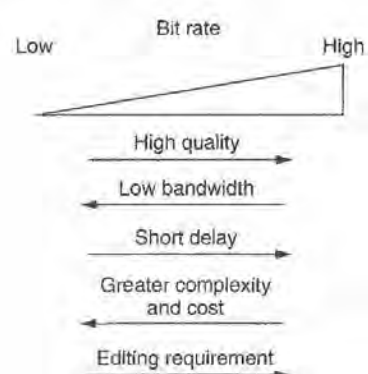


Figure 7.2 There are a large number of conflicting demands in the design of a compression system. Here the effect of a number of choices is shown to alter the balance between economy and bandwidth.

Google Exhibit No. 1005
Page 121

Google Exhibit No. 1063
Page 0376

received with a time shift. In this case additional buffering is needed at both ends of the channel so that the variable delay characteristic of the channel is invisible to the MPEG layer.

There are now many members of the MPEG family and it can be difficult to decide which type of MPEG coding to use. Figure 7.2 shows some general guidelines. As a decoder becomes more complex, the bit rate which can be used for a given quality will fall. However, this will be accompanied by an increase in power consumption at the encoder and decoder as well as an increased delay. The designer will have to balance the cost of the bandwidth with the cost of the encoding and decoding equipment.

7.2 Video phones

The telephone is a classic example of limited bandwidth. If there is a market for video phones, then compression is the only way forward. The facial animation coding developed for MPEG-4 is an obvious contender as it allows a reasonable reproduction of the caller's face with a data rate of only a few kilobits per second. This is considerably less than the data rate needed for the speech. Video telephony is relatively easy between desk-based equipment, where the camera can be mounted on top of the display, but it is not clear how this can be implemented in a cellular telephone which is held against the user's ear. Another important issue for cellular telephones is the amount of signal processing needed to decode and display facial animation data. Essentially the decoder is a real-time rendering engine and this is only practicable in a portable product if the current consumption is within the range of available battery technology.

7.3 Digital television broadcasting

Digital television broadcasting relies on the combination of a number of fundamental technologies. These are: MPEG-2 compression to reduce the bit rate, multiplexing to combine picture and sound data into a common bitstream, digital modulation schemes to reduce the RF bandwidth needed by a given bit rate and error correction to reduce the error statistics of the channel down to a value acceptable to MPEG data.

MPEG is a compression and multiplexing standard and does not specify how error correction should be performed. Consequently a transmission standard must define a system which has to correct essentially all errors such that the delivery mechanism is transparent.

Google Exhibit No. 1005
Page 122

Google Exhibit No. 1063
Page 0377

DVB (digital video broadcasting)[1] is a standard which incorporates MPEG-2 picture and sound coding but which also specifies all the additional steps needed to deliver an MPEG transport stream from one place to another. This transport stream will consist of a number of elementary streams of video and audio, where the audio may be coded according to MPEG audio standard or AC-3. In a system working within its capabilities, the picture and sound quality will be determined only by the performance of the compression system and not by the RF transmission channel. This is the fundamental difference between analog and digital broadcasting. In analog television broadcasting, the picture quality may be limited by composite video encoding artifacts as well as transmission artifacts such as noise and ghosting. In digital television broadcasting the picture quality is determined instead by the compression artifacts and interlace artifacts if interlace has been retained.

If the received error rate increases for any reason, once the correcting power is used up, the system will degrade rapidly as uncorrected errors enter the MPEG decoder. In practice decoders will be programmed to recognize the condition and blank or mute to avoid outputting garbage. As a result digital receivers tend either to work well or not at all.

It is important to realize that the signal strength in a digital system does not translate directly to picture quality. A poor signal will increase the number of bit errors. Provided that this is within the capability of the error-correction system, there is no visible loss of quality. In contrast, a very powerful signal may be unusable because of similarly powerful reflections due to multipath propagation.

Whilst in one sense an MPEG transport stream is only data, it differs from generic data in that it must be presented to the viewer at a particular rate. Generic data are usually asynchronous, whereas baseband video and audio are synchronous. However, after compression and multiplexing audio and video are no longer precisely synchronous and so the term *isochronous* is used. This means a signal which was at one time synchronous and will be displayed synchronously, but which uses buffering at transmitter and receiver to accommodate moderate timing errors in the transmission.

Clearly another mechanism is needed so that the time axis of the original signal can be re-created on reception. The time stamp and program clock reference system of MPEG does this.

Figure 7.3 shows that the concepts involved in digital television broadcasting exist at various levels which have an independence not found in analog technology. In a given configuration a transmitter can radiate a given payload data bit rate. This represents the useful bit rate and does not include the necessary overheads needed by error correction, multiplexing or synchronizing. It is fundamental that the transmission system does not care what this payload bit rate is used for. The entire

Google Exhibit No. 1005
Page 123

Google Exhibit No. 1063
Page 0378

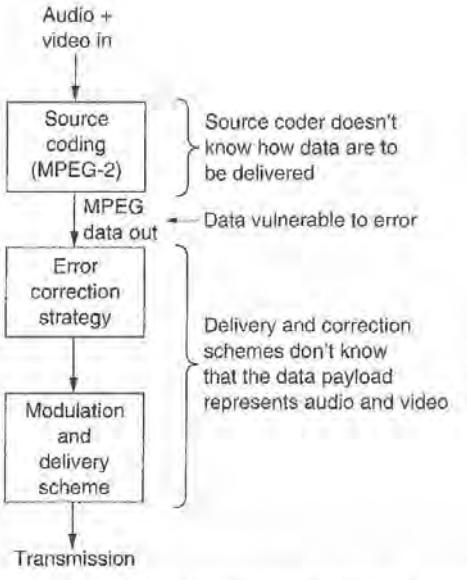


Figure 7.3 Source coder doesn't know delivery mechanism and delivery doesn't need to know what the data mean.

capacity may be used up by one high-definition channel, or a large number of heavily compressed channels may be carried. The details of this data usage are the domain of the *transport stream*. The multiplexing of transport streams is defined by the MPEG standards, but these do not define any error correction or transmission technique.

At the lowest level in Figure 7.3 is the source coding scheme, in this case MPEG-2 compression results in one or more elementary streams, each of which carries a video or audio channel. Figure 7.4 shows that elementary streams are multiplexed into a transport stream. The viewer then selects the desired elementary stream from the transport stream. Metadata in the transport stream ensure that when a video elementary stream is chosen, the appropriate audio elementary stream will automatically be selected.

A key difference between analog and digital transmission is that the transmitter output is switched between a number of discrete states rather than continuously varying. The process is called channel coding which is the digital equivalent of modulation.[2] A good code minimizes the channel bandwidth needed for a given bit rate. This quality of the code is measured in bits/s/Hz.

Where the SNR is poor, as in satellite broadcasting, the amplitude of the signal will be unstable, and phase modulation is used. Figure 7.5 shows that phase-shift keying (PSK) can use two or more phases. When four

Google Exhibit No. 1005
Page 124

Google Exhibit No. 1063
Page 0379

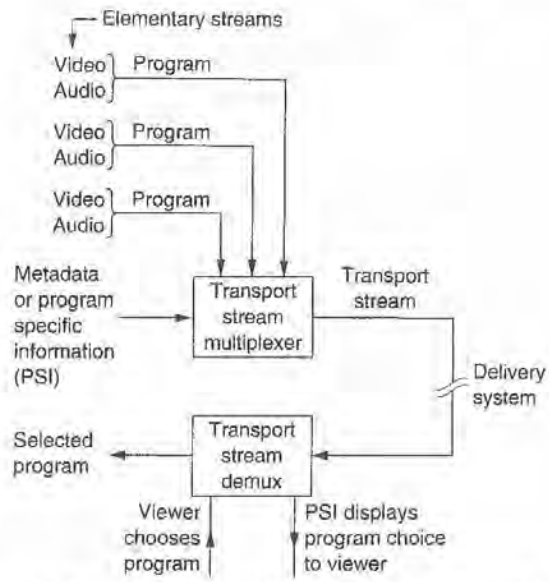
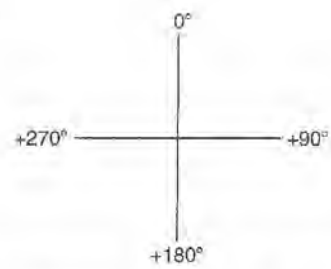


Figure 7.4 Program Specific Information helps the demultiplexer to select the required program.



Data	Phase shift
0 0	0°
0 1	90°
1 0	270°
1 1	180°

Figure 7.5 Differential quadrature phase shift keying (DQPSK).

phases in quadrature are used, the result is quadrature phase shift keying or QPSK. Each period of the transmitted waveform can have one of four phases and therefore conveys the value of two data bits. 8-PSK uses eight phases and can carry three bits per symbol where the SNR is adequate. PSK is generally encoded in such a way that a knowledge of absolute phase is not needed at the receiver. Instead of encoding the signal phase directly, the data determine the magnitude of the phase shift between symbols.

In terrestrial transmission more power is available than, for example, from a satellite and so a stronger signal can be delivered to the receiver. Where a better SNR exists, an increase in data rate can be had using

Google Exhibit No. 1005
Page 125

Google Exhibit No. 1063
Page 0380

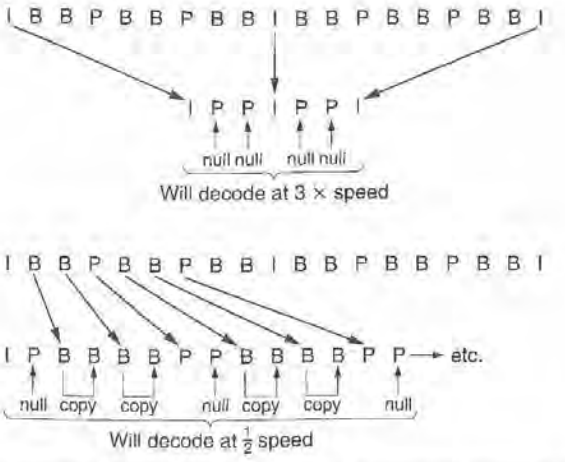


Figure 7.21 By manipulating the elementary stream, a compliant bitstream can be produced which will play at a different speed from the original.

discarding some of the pictures in the group. This has to be done with caution as the pictures are heavily interdependent. However, if the group is terminated after a *P* picture, all pictures up to that point are decodable. The resultant motion portrayal will not be smooth because pictures are being omitted, but the goal is simply to aid the viewer in finding the right place in the recording so this is of no consequence. In practice the results are good, not least because the noise bar of the analog VCR is eliminated.

High forward speeds can be obtained by selecting only the *I* pictures from the recorded bitstream. These cannot be transmitted to the decoder directly, because the *I* pictures contain a lot of data and the decoder buffer might overflow. Consequently the data rate is diluted by sending null *P* pictures. These are *P* pictures in which all of the vectors are zero and the residual data are also zero. Null *P* pictures require very few data but have the effect of converting an *I* picture into another identical picture. A similar technique can be used to decode in reverse. With a normal MPEG bidirectionally coded bitstream, reverse decoding is impossible, but by taking *I* pictures in reverse order and padding them with null-*P* pictures a reverse replay is obtained.

7.7 Networks

A network is basically a communication resource which is shared for economic reasons. Like any shared resource, decisions have to be made somewhere and somehow about how the resource is to be used. In the

Google Exhibit No. 1005
Page 126

Google Exhibit No. 1063
Page 0381

absence of such decisions the resultant chaos will be such that the resource might as well not exist.

In communications networks the resource is the ability to convey data from any node or port to any other. On a particular cable, clearly only one transaction of this kind can take place at any one instant even though in practice many nodes will simultaneously be wanting to transmit data. Arbitration is needed to determine which node is allowed to transmit.

Data networks originated to serve the requirements of computers and it is a simple fact that most computer processes don't need to be performed in real time or indeed at a particular time at all. Networks tend to reflect that background as many of them, particularly the older ones, are asynchronous. Asynchronous means that the time taken to deliver a given quantity of data is unknown. A TDM system may chop the data into several different transfers and each transfer may experience delay according to what other transfers the system is engaged in. Ethernet and most storage system buses are asynchronous. For broadcasting purposes an asynchronous delivery system is no use at all, but for copying an MPEG data file between two storage devices an asynchronous system is perfectly adequate.

The opposite extreme is the synchronous system in which the network can guarantee a constant delivery rate and a fixed and minor delay. An AES/EBU digital audio router or an SDI digital video router is a synchronous network. In between asynchronous and synchronous networks reside the isochronous approaches which cause a fixed moderate delay. These can be thought of as sloppy synchronous networks or more rigidly controlled asynchronous networks.

These three different approaches are needed for economic reasons. Asynchronous systems are very efficient because as soon as one transfer completes another one can begin. This can only be achieved by making every device wait with its data in a buffer so that transfer can start immediately. Asynchronous systems also make it possible for low bit rate devices to share a network with high bit rate devices. The low bit rate device will only need a small buffer and will send few cells, whereas the high bit rate device will send more cells.

Isochronous systems try to give the best of both worlds, generally by sacrificing some flexibility in block size. Modern networks are tending to be part isochronous and part asynchronous so that the advantages of both are available.

There are a number of different arbitration protocols and these have evolved to support the needs of different types of network. In small networks, such as LANs, a single point failure which halts the entire network may be acceptable, whereas in a public transport network owned by a telecommunications company, the network will be redundant so that if a particular link fails data may be sent via an alternative route.

Google Exhibit No. 1005
Page 127

Google Exhibit No. 1063
Page 0382

A link which has reached its maximum capacity may also be supplanted by transmission over alternative routes.

In physically small networks, arbitration may be carried out in a single location. This is fast and efficient, but if the arbitrator fails it leaves the system completely crippled. The processor buses in computers work in this way.

In centrally arbitrated systems the arbitrator needs to know the structure of the system and the status of all the nodes. Following a configuration change, due perhaps to the installation of new equipment, the arbitrator needs to be told what the new configuration is, or have a mechanism which allows it to explore the network and learn the configuration. Central arbitration is only suitable for small networks which change their configuration infrequently.

In other networks the arbitration is distributed so that some decision-making ability exists in every node. This is less efficient but is does allow at least some of the network to continue operating after a component failure.

Distributed arbitration also means that each node is self-sufficient and so no changes need to be made if the network is reconfigured by adding or deleting a node. This is the only possible approach in wide area networks where the structure may be very complex and changes dynamically in the event of failures or overload.

Ethernet uses distributed arbitration. FireWire is capable of using both types of arbitration. A small amount of decision-making ability is built into every node so that distributed arbitration is possible. However, if one of the nodes happens to be a computer, it can run a centralized arbitration algorithm.

The physical structure of a network is subject to some variation as Figure 7.22 shows. In radial networks, (a), each port has an unique cable connection to a device called a *hub*. The hub must have one connection for every port and this limits the number of ports. However, a cable failure will only result in the loss of one port. In a ring system (b) the nodes are connected like a daisy chain with each node acting as a feedthrough. In this case the arbitration requirement must be distributed. With some protocols, a single cable break doesn't stop the network operating. Depending on the protocol, simultaneous transactions may be possible provided they don't require the same cable. For example, in a storage network a disk drive may be outputting data to an editor whilst another drive is backing up data to a tape streamer. For the lowest cost, all nodes are physically connected in parallel to the same cable. Figure 7.22(c) shows that a cable break would divide the network into two halves, but it is possible that the impedance mismatch at the break could stop both halves working.

One of the concepts involved in arbitration is priority which is fundamental to providing an appropriate quality of service. If two

Google Exhibit No. 1005
Page 128

Google Exhibit No. 1063
Page 0383

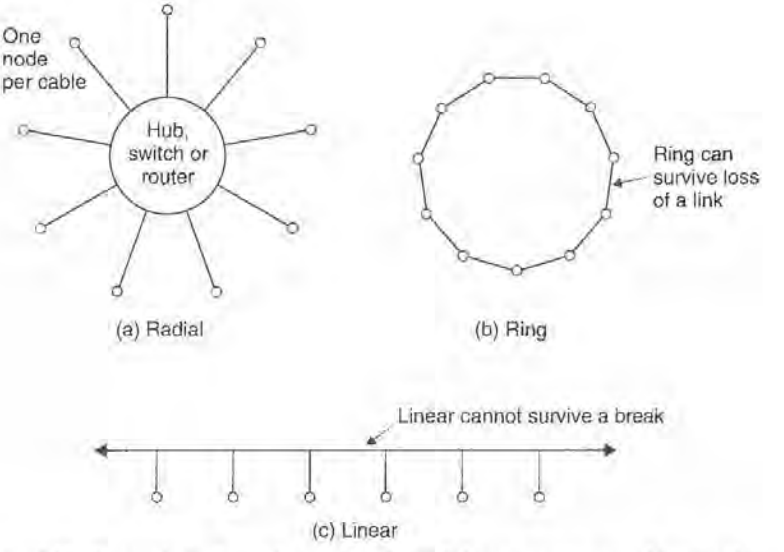


Figure 7.22 Network configurations. At (a) the radial system uses one cable to each node. (b) Ring system uses less cable than radial. (c) Linear system is simple but has no redundancy.

processes both want to use a network, the one with the highest priority would normally go first.

Attributing priority must be done carefully because some of the results are non-intuitive. For example, it may be beneficial to give a high priority to a humble device which has a low data rate for the simple reason that if it is given use of the network it won't need it for long. In a television environment transactions concerned with on-air processes would have priority over file transfers concerning production and editing.

When a device gains access to the network to perform a transaction, generally no other transaction can take place until it has finished. Consequently it is important to limit the amount of time that a given port can stay on the bus. In this way when the time limit expires, a further arbitration must take place. The result is that the network resource rotates between transactions rather than one transfer hogging the resource and shutting everyone else out. It follows from the presence of a time (or data quantity) limit that ports must have the means to break large files up into frames or cells and reassemble them on reception. This process is sometimes called *adaptation*. If the data to be sent originally exist at a fixed bit rate, some buffering will be needed so that the data can be time-compressed into the available frames. Each frame must be contiguously numbered and the system must transmit a file size or word count so that the receiving node knows when it has received every frame in the file.

Google Exhibit No. 1005
Page 129

Google Exhibit No. 1063
Page 0384

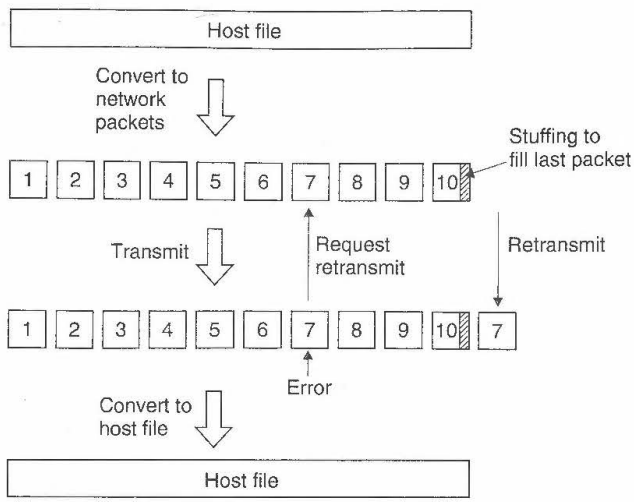


Figure 7.23 Receiving a file which has been divided into packets allows for the retransmission of just the packet in error.

The error-detection system interacts with this process because if any frame is in error on reception, the receiving node can ask for a retransmission of the frame. This is more efficient than retransmitting the whole file. Figure 7.23 shows the mechanism of retransmission where one packet has been lost or is in error. The retransmitted packet is inserted at the correct place in the bitstream by the receiver.

Breaking files into frames helps to keep down the delay experienced by each process using the network. Figure 7.24 shows a frame-multiplexed transmission. Each frame may be stored ready for transmission in a silo memory. It is possible to make the priority a function of the number of frames in the silo, as this is a direct measure of how long a process has been kept waiting. Isochronous systems must do this in order to meet maximum delay specifications.

When delivering an MPEG bitstream over an isochronous network, the network transmitter and receiver will require buffering over and above that provided in the MPEG encoder and decoder. This will enable the receiver to re-create a bitstream at the decoder which has essentially the same temporal characteristics as that leaving the encoder, except for a fixed delay. The greater the maximum delay of the network, the greater that fixed delay will have to be. Providing buffering is expensive and delays cause difficulty in applications such as videoconferencing. In this case it is better to use a network protocol which can limit delay by guaranteeing bandwidth.

A central arbitrator is relatively simple to implement because when all decisions are taken centrally there can be no timing difficulty (assuming

Google Exhibit No. 1005
Page 130

Google Exhibit No. 1063
Page 0385

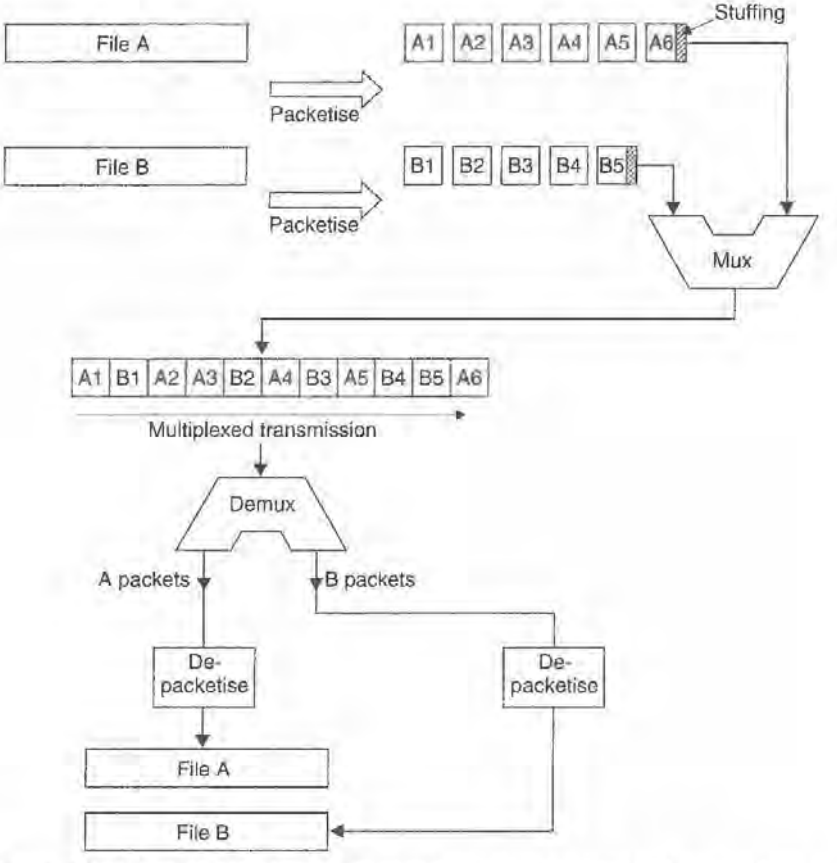


Figure 7.24 Files are broken into frames or packets for multiplexing with packets from other users. Short packets minimize the time between the arrival of successive packets. The priority of the multiplexing must favour isochronous data over asynchronous data.

a well-engineered system). In a distributed system, there is an extra difficulty due to the finite time taken for signals to travel down the data paths between nodes.

Figure 7.25 shows the structure of Ethernet which uses a protocol called CSMA/CD (carrier sense multiple access with collision detect) developed by DEC and Xerox. This is a distributed arbitration network where each node follows some simple rules. The first of these is not to transmit if an existing bus signal is detected. The second is not to transmit more than a certain quantity of data before releasing the bus. Devices wanting to use the bus will see bus signals and so will wait until the present bus transaction finishes. This must happen at some point because of the frame size limit. When the frame is completed, signalling on the bus should cease. The first device to sense the bus becoming free and to assert its own

Google Exhibit No. 1005
Page 131

Google Exhibit No. 1063
Page 0386

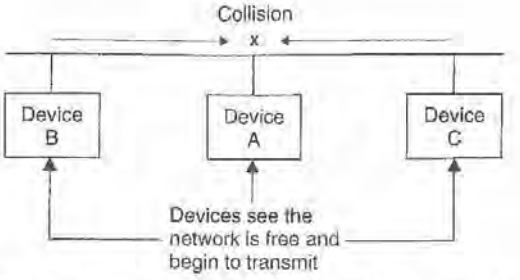


Figure 7.25 In Ethernet collisions can occur because of the finite speed of the signals. A 'back-off' algorithm handles collisions, but they do reduce the network throughput.

signal will prevent any other nodes transmitting according to the first rule. Where numerous devices are present it is possible to give them a priority structure by providing a delay between sensing the bus coming free and beginning a transaction. High-priority devices will have a short delay so they get in first. Lower-priority devices will only be able to start a transaction if the high-priority devices don't need to transfer.

It might be thought that these rules would be enough and everything would be fine. Unfortunately the finite signal speed means that there is a flaw in the system. Figure 7.25 shows why. Device A is transmitting and devices B and C both want to transmit and have equal priority. At the end of A's transaction, devices B and C see the bus become free at the same instant and start a transaction. With two devices driving the bus, the resultant waveform is meaningless. This is known as a collision and all nodes must have means to recover from it. First, each node will read the bus signal at all times. When a node drives the bus, it will also read back the bus signal and compare it with what was sent. Clearly if the two are the same all is well, but if there is a difference, this must be because a collision has occurred and two devices are trying to determine the bus voltage at once.

If a collision is detected, both colliding devices will sense the disparity between the transmitted and readback signals, and both will release the bus to terminate the collision. However, there is no point is adhering to the simple protocol to reconnect because this will simply result in another collision. Instead each device has a built-in delay which must expire before another attempt is made to transmit. This delay is not fixed, but is controlled by a random number generator and so changes from transaction to transaction.

The probability of two node devices arriving at the same delay is infinitesimally small. Consequently if a collision does occur, both devices will drop the bus, and they will start their back-off timers. When the first timer expires, that device will transmit and the other will see the

Google Exhibit No. 1005
Page 132

Google Exhibit No. 1063
Page 0387

transmission and remain silent. In this way the collision is not only handled, but is prevented from happening again.

The performance of Ethernet is usually specified in terms of the bit rate at which the cabling runs. However, this rate is academic because it is not available all of the time. In a real network bit rate is lost by the need to send headers and error-correction codes and by the loss of time due to interframe spaces and collision handling. As the demand goes up, the number of collisions increases and throughput goes down. Collision-based arbitrators do not handle congestion well.

An alternative method of arbitration developed by IBM is shown in Figure 7.26. This is known as a *token ring* system. All the nodes have an

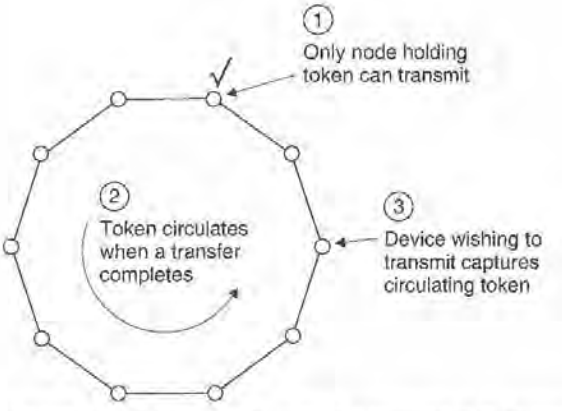


Figure 7.26 In a token ring system only the node in possession of the token can transmit so collisions are impossible. In very large rings the token circulation time causes loss of throughput.

input and an output and are connected in a ring which must be complete for the system to work. Data circulate in one direction only. If data are not addressed to a node which receives it, the data will be passed on. When the data arrive at the addressed node, that node will capture the data as well as passing them on with an acknowledge added. Thus the data packet travels right around the ring back to the sending node. When the sending node receives the acknowledge, it will transmit a token packet. This token packet passes to the next node, which will pass it on if it does not wish to transmit. If no device wishes to transmit, the token will circulate endlessly. However, if a device has data to send, it simply waits until the token arrives again and captures it. This node can now transmit data in the knowledge that there cannot be a collision because no other node has the token.

Google Exhibit No. 1005
Page 133

Google Exhibit No. 1063
Page 0388

In simple token ring systems, the transmitting node transmits idle characters after the data packet has been sent in order to maintain synchronization. The idle character transmission will continue until the acknowledge arrives. In the case of long packets the acknowledge will arrive before the packet has all been sent and no idle characters are necessary. However, with short packets idle characters will be generated. These idle characters use up ring bandwidth.

Later token ring systems use *early token release* (ETR). After the packet has been transmitted, the sending node sends a token straight away. Another node wishing to transmit can do so as soon as the current packet has passed. It might be thought that the nodes on the ring would transmit in their physical order, but this is not the case because a priority system exists. Each node can have a different priority if necessary. If a high-priority node wishes to transmit, as a packet from elsewhere passes through that node, the node will set *reservation bits* with its own priority level. When the sending node finishes and tranmits a token, it will copy that priority level into the token.

In this way nodes with a lower priority level will pass the token on instead of capturing it. The token will ultimately arrive at the high-priority node.

The token ring system has the advantage that it does not waste throughput with collisions and so the full capacity is always available. However, if the ring is broken the entire network fails.

In Ethernet the performance is degraded by the number of transactions, not the number of nodes, whereas in token ring the performance is degraded by the number of nodes.

7.8 FireWire

FireWire[3] is actually an Apple Computers Inc. trade name for the interface which is formally known as IEEE 1394–1995. It was originally intended as a digital audio network, but grew out of recognition. FireWire is more than just an interface as it can be used to form networks and if used with a computer effectively extends the computer's data bus. Figure 7.27 shows that devices are simply connected together as any combination of daisy-chain or star network.

Any pair of devices can communicate in either direction, and arbitration ensures that only one device transmits at once. Intermediate devices simply pass on transmissions. This can continue even if the intermediate device is powered down as the FireWire carries power to keep repeated functions active.

Communications are divided into *cycles* which have a period of 125 μs. During a cycle, there are 64 time slots. During each time slot, any one

Google Exhibit No. 1005
Page 134

Google Exhibit No. 1063
Page 0389

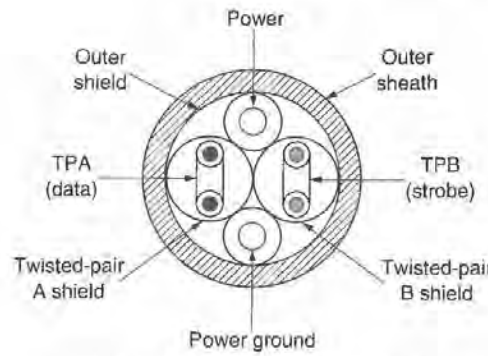


Figure 7.28 FireWire uses twin twisted pairs and a power pair.

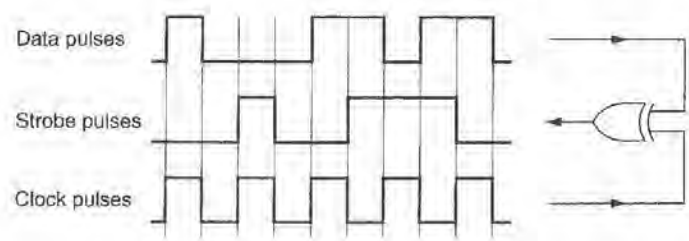


Figure 7.29 The strobe signal is the X-OR of the data and the bit clock. The data and strobe signals together form a self-clocking system.

carries a signal called *strobe*, which is the exclusive-OR of the data and the clock. Thus whenever a run of identical bits results in no transitions in the data, the strobe signal will carry transitions. At the receiver another exclusive-OR gate adds data and strobe to re-create the clock.

This signalling technique is subject to skew between the two twisted pairs and this limits cable lengths to about 10 metres between nodes. Thus FireWire is not a long-distance interface technique, instead it is very useful for interconnecting a large number of devices in close proximity. Using a copper interconnect, FireWire can run at 100, 200 or 400 megabits/s, depending on the specific hardware. It is proposed to create an optical fibre version which would run at gigabit speeds.

7.9 Broadband networks and ATM

Broadband ISDN (B-ISDN) is the successor to N-ISDN and in addition to providing more bandwidth, offers practical solutions to the delivery of any conceivable type of data. The flexibility with which ATM operates means that intermittent or one-off data transactions which only require

Google Exhibit No. 1005
Page 135

Google Exhibit No. 1063
Page 0390

asynchronous delivery can take place alongside isochronous MPEG video delivery. This is known as *application independence* whereby the sophistication of isochronous delivery does not raise the cost of asynchronous data. In this way, generic data, video, speech and combinations of the above can co-exist.

ATM is multiplexed, but it is not time-division multiplexed. TDM is inefficient because if a transaction does not fill its allotted bandwidth, the capacity is wasted. ATM does not offer fixed blocks of bandwidth, but allows infinitely variable bandwidth to each transaction. This is done by converting all host data into small fixed size cells at the adaptation layer. The greater the bandwidth needed by a transaction, the more cells per second are allocated to that transaction. This approach is superior to the fixed bandwidth approach, because if the bit rate of a particular transaction falls, the cells released can be used for other transactions so that the full bandwidth is always available.

As all cells are identical in size, a multiplexer can assemble cells from many transactions in an arbitrary order. The exact order is determined by the quality of service required, where the time positioning of isochronous data would be determined first, with asynchronous data filling the gaps.

Figure 7.30 shows how a broadband system might be implemented. The transport network would typically be optical fibre based, using SONET (Synchronous optical network) or SDH (synchronous digital hierarchy). These standards differ in minor respects. Figure 7.31 shows

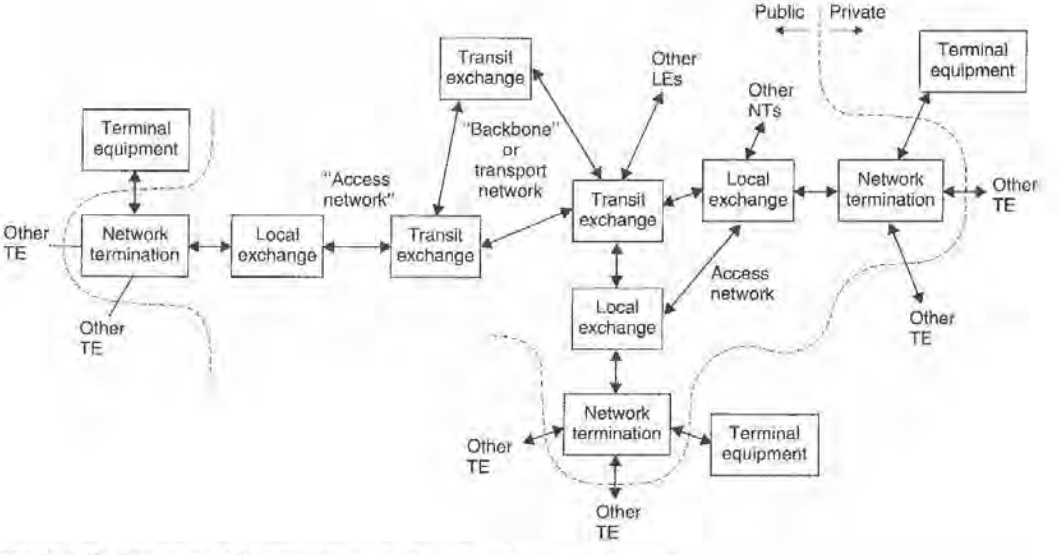


Figure 7.30 Structure and terminology of a broadband network. See text.

Google Exhibit No. 1005
Page 136

Google Exhibit No. 1063
Page 0391

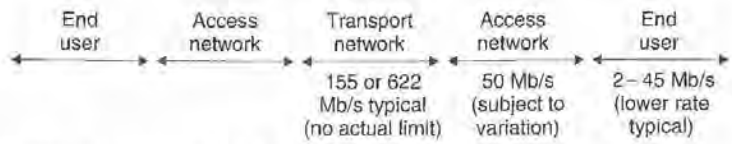


Figure 7.31 Bit rates available in SONET and SDH.

the bit rates available in each. Lower bit rates will be used in the access networks which will use different technology such as xDSL.

SONET and SDH assemble ATM cells into a structure known as a *container* in the interests of efficiency. Containers are passed intact between exchanges in the transport network. The cells in a container need not belong to the same transaction, they simply need to be going the same way for at least one transport network leg.

The cell-routing mechanism of ATM is unusual and deserves explanation. In conventional networks, a packet must carry the complete destination address so that at every exchange it can be routed closer to its destination. The exact route by which the packet travels cannot be anticipated and successive packets in the same transaction may take different routes. This is known as a *connectionless* protocol.

In contrast, ATM is a *connection oriented* protocol. Before data can be transferred, the network must set up an end-to-end route. Once this is done, the ATM cells do not need to carry a complete destination address. Instead they only need to carry enough addressing so that an exchange or switch can distinguish between all of the expected transactions.

The end-to-end route is known as a *virtual channel* which consists of a series of *virtual links* between switches. The term virtual channel is used because the system acts like a dedicated channel even though physically it is not. When the transaction is completed the route can be dismantled so that the bandwidth is freed for other users. In some cases, such as delivery of a TV station's output to a transmitter, or as a replacement for analog cable TV the route can be set up continuously to form what is known as a *permanent virtual channel*.

The addressing in the cells ensures that all cells with the same address take the same path, but owing to the multiplexed nature of ATM, at other times and with other cells a completely different routing scheme may exist. Thus the routing structure for a particular transaction always passes cells by the same route, but the next cell may belong to another transaction and will have a different address causing it to routed in another way.

The addressing structure is hierarchical. Figure 7.32(a) shows the ATM cell and its header. The cell address is divided into two fields, the virtual channel identifier and the virtual path identifier. Virtual paths

Google Exhibit No. 1005
Page 137

Google Exhibit No. 1063
Page 0392

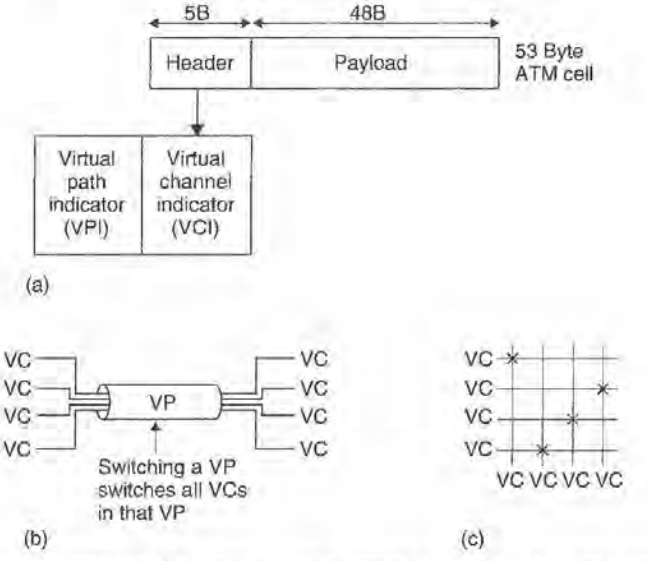


Figure 7.32 The ATM cell (a) carries routing information in the header. ATM paths carrying a group of channels can be switched in a virtual path switch (b). Individual channel switching requires a virtual channel switch which is more complex and causes more delay.

are logical groups of virtual channels which happen to be going the same way. An example would be the output of a video-on-demand server travelling to the first switch. The virtual path concept is useful because all cells in the same virtual path can share the same container in a transport network. A virtual path switch shown in Figure 7.32(b) can operate at the container level whereas a virtual channel switch (c) would need to dismantle and reassemble containers.

When a route is set up, at each switch a table is created. When a cell is received at a switch the VPI and/or VCI code is looked up in the table and used for two purposes. First, the configuration of the switch is obtained, so that this switch will correctly route the cell. Second, the VPI and/or VCI codes may be updated so that they correctly control the next switch. This process repeats until the cell arrives at its destination.

In order to set up a path, the initiating device will initially send cells containing an ATM destination address, the bandwidth and quality of service required. The first switch will reply with a message containing the VPI/VCI codes which are to be used for this channel. The message from the initiator will propagate to the destination, creating look-up tables in each switch. At each switch the logic will add the requested bandwidth to the existing bandwidth in use to check that the requested

Google Exhibit No. 1005
Page 138

Google Exhibit No. 1063
Page 0393

quality of service can be met. If this succeeds for the whole channel, the destination will reply with a connect message which propagates back to the initiating device as confirmation that the channel has been set up. The connect message contains an unique call reference value which identifies this transaction. This is necessary because an initiator such a file server may be initiating many channels and the connect messages will not necessarily return in the same order as the set-up messages were sent. The last switch will confirm receipt of the connect message to the destination and the initiating device will confirm receipt of the connect message to the first switch.

7.10 ATM AALs

ATM works by dividing all real data messages into cells of 48 bytes each. At the receiving end, the original message must be re-created. This can take many forms and Figure 7.33 shows some possibilities. The message may be a generic data file having no implied timing structure, a serial bitstream with a fixed clock frequency, known as UDT (unstructured data transfer) or a burst of data bytes from a TDM system.

The application layer in ATM has two sub-layers shown in Figure 7.34. The first is the segmentation and reassembly (SAR) sub-layer

Generic data file having no timebase
Constant bit rate serial data stream
Audio/video data requiring a timebase
Compressed A/V data with fixed bit rate
Compressed A/V data with variable bit rate

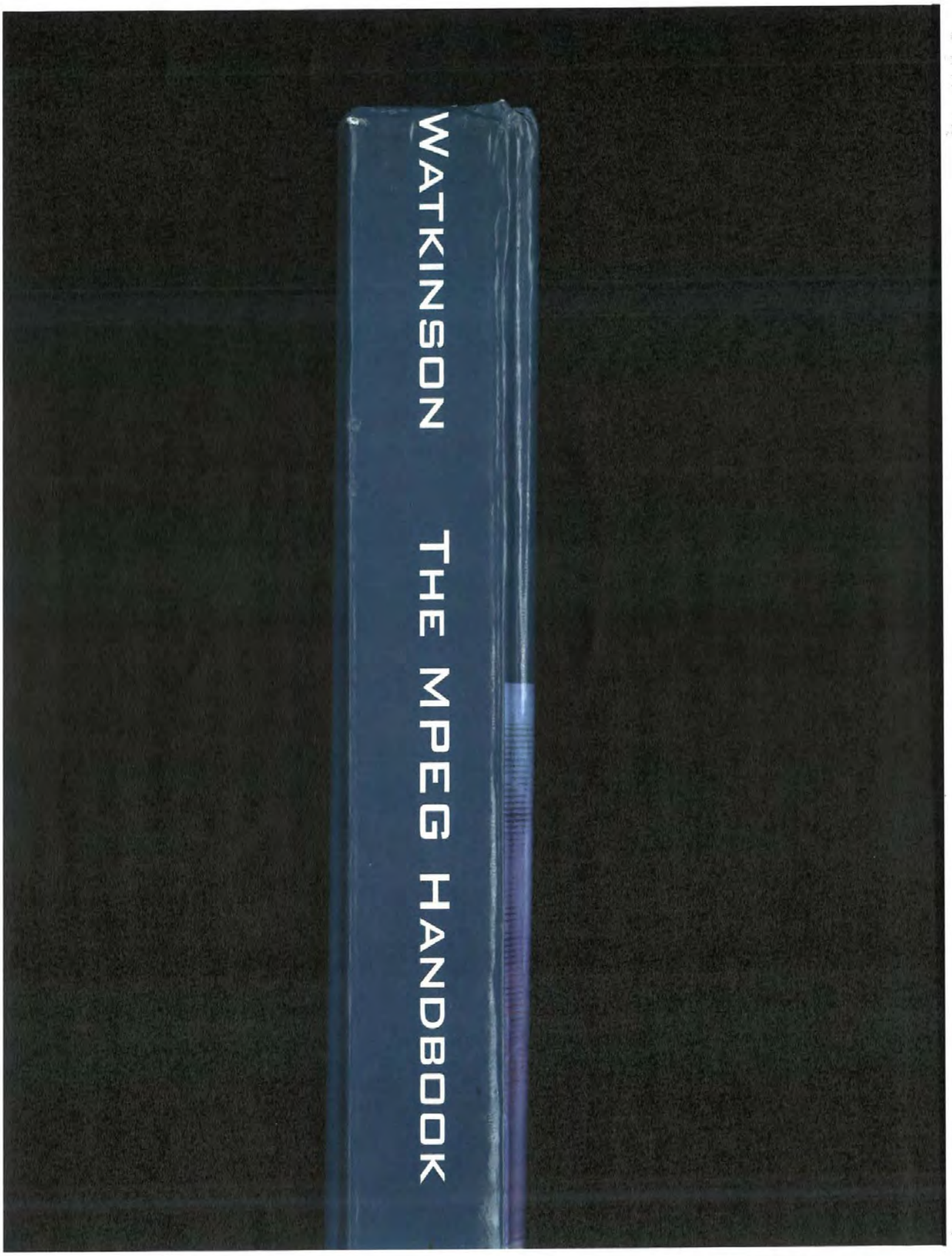
Figure 7.33 Types of data which may need adapting to ATM.

ATM Application Layer	Convergence sublayer	Recovers timing of original data
	Segmentation and reassembly	Divides data into cells for transport Reassembles original data format

Figure 7.34 ATM adaption layer has two sublayers, segmentation and convergence.

Google Exhibit No. 1005
Page 139

Google Exhibit No. 1063
Page 0394


WATKINSON
THE MPEG HANDBOOK

BROADCASTING AND COMMUNICATIONS

- Benefit from John Watkinson's in-depth understanding of MPEG technology
- Comprehensive coverage of the main and latest standards -- MPEG-1, MPEG-2, MPEG-4
- Learn how you can apply MPEG technology to achieve the best results

Review of MPEG-2

'... a thorough and very lucid introduction to audio and video compression. Through the use of simple, clear diagrams, and some of the clearest technical writing I have come across, Watkinson makes the whole process look remarkably straightforward... an absorbing read for the curious, and a useful source of hard information for those with a professional interest.' ***Electronics Australia***

A complete, professional 'bible' on all aspects of audio and video compression using MPEG technology, including the MPEG-4 standard. *The MPEG Handbook* is both a theoretical and practical treatment of the subject. Fundamental knowledge is provided alongside practical guidance on how to avoid quality loss for example, and the creation of MPEG elementary streams and their multiplexing into transport streams (including the problems involved in synchronizing all of the signals in a multiplex).

Previously titled *MPEG-2*, the book has now been extensively revised to cover all the latest standards, including a comprehensive and clear description of the MPEG-4 standard and the latest applications of the technology, such as the delivery of compressed audio, video, graphics and interactive capability over the Internet and broadband networks. John Watkinson also gives an invaluable treatment of significant new extensions made available by MPEG-4, such as object coding, mesh coding and animation.

The clarity of explanation and depth of technical detail combine to make this book an essential and definitive reference work.

John Watkinson is an international consultant in audio, video and data recording. He is a Fellow of the AES, a member of the British Computer Society and a chartered information systems practitioner. He presents lectures, seminars, conference papers and training courses worldwide. He is the author of many other Focal Press books, including: *The Art of Digital Video, An Introduction to Digital Video, The Art of Digital Audio, An Introduction to Digital Audio, Convergence in Broadcast and Communications Media, MPEG-2, Television Fundamentals* and *Audio for Television.*

Focal Press



Ex. 1005A

LIBRARY | CATALOG

1 of 1

BOOK

The MPEG handbook : MPEG-1, MPEG-2, MPEG-4

Full Record MARC Tags

Personal name

Watkinson, John, 1950-

Main title

The MPEG handbook : MPEG-1, MPEG-2, MPEG-4 / John Watkinson.

Published/Created

Oxford [England] ; Boston : Focal Press, 2001.

Request this Item | LC Find It

Links

Links

Publisher description http://www.loc.gov/catdir/description/els031/2001044789.html

Table of contents http://www.loc.gov/catdir/toc/els031/2001044789.html

More Information

LCCN Permalink

https://lccn.loc.gov/2001044789

Google Exhibit No. 1063
Page 0398

Description	ix, 395 p. : ill. ; 26 cm.
ISBN	0240516567 (alk. paper)
LC classification	TK6680.5 .W3843 2001
LC Subjects	Video compression--Handbooks, manuals, etc. MPEG (Video coding standard)--Handbooks, manuals, etc.
Browse by shelf order	TK6680.5
Notes	Includes bibliographical references and index.
LCCN	2001044789
Dewey class no.	621.388
Type of material	Book

Item Availability

CALL NUMBER	TK6680.5 .W3843 2001 Copy 1
Request in	Jefferson or Adams Building Reading Rooms
Status	Not Charged

Google Exhibit No. 1063
Page 0399

LIBRARY | CATALOG

1 of 1

BOOK

The MPEG handbook : MPEG-1, MPEG-2, MPEG-4

Full Record **MARC Tags**

000		01246cam a22002774a 4500
001		12515198
005		20030709193435.0
008		010827s2001 enka bf 001 0 eng
906	__	|a 7 |b cbc |c orignew |d 1 |e ocip |f 20 |g y-gencatlg
925	0_	|a acquire |b 1 shelf copy |x policy default
955	__	|a pc20 2001-08-27 to ASCD |c jf07 2001-08-28 to subject |d jf04 2001-08-28 to S.L. | e jf25 2001-08-31 to Dewey |a aa07 2001-09-04 |a ps16 2001-11-07 bk rec'd, to CIP ver.; | f pv04 2001-11-21 CIP ver to BCCD
010	__	|a 2001044789
020	__	|a 0240516567 (alk. paper)
040	__	|a DLC |c DLC |d DLC
042	__	|a pcc
050	00	|a TK6680.5 |b .W3843 2001
082	00	|a 621.388 |2 21
100	1_	|a Watkinson, John, |d 1950-
245	14	|a The MPEG handbook : |b MPEG-1, MPEG-2, MPEG-4 / |c John Watkinson.
260	__	|a Oxford [England] ; |a Boston : |b Focal Press, |c 2001.
300	__	|a ix, 395 p. : |b ill. ; |c 26 cm.
504	__	|a Includes bibliographical references and index.
650	_0	|a Video compression |v Handbooks, manuals, etc.
650	_0	|a MPEG (Video coding standard) |v Handbooks, manuals, etc.

Google Exhibit No. 1063
Page 0400

856	42	|3 Publisher description |u http://www.loc.gov/catdir/description/els031/2001044789.html
856	41	|3 Table of contents |u http://www.loc.gov/catdir/toc/els031/2001044789.html

Request this Item | LC Find It

Item Availability

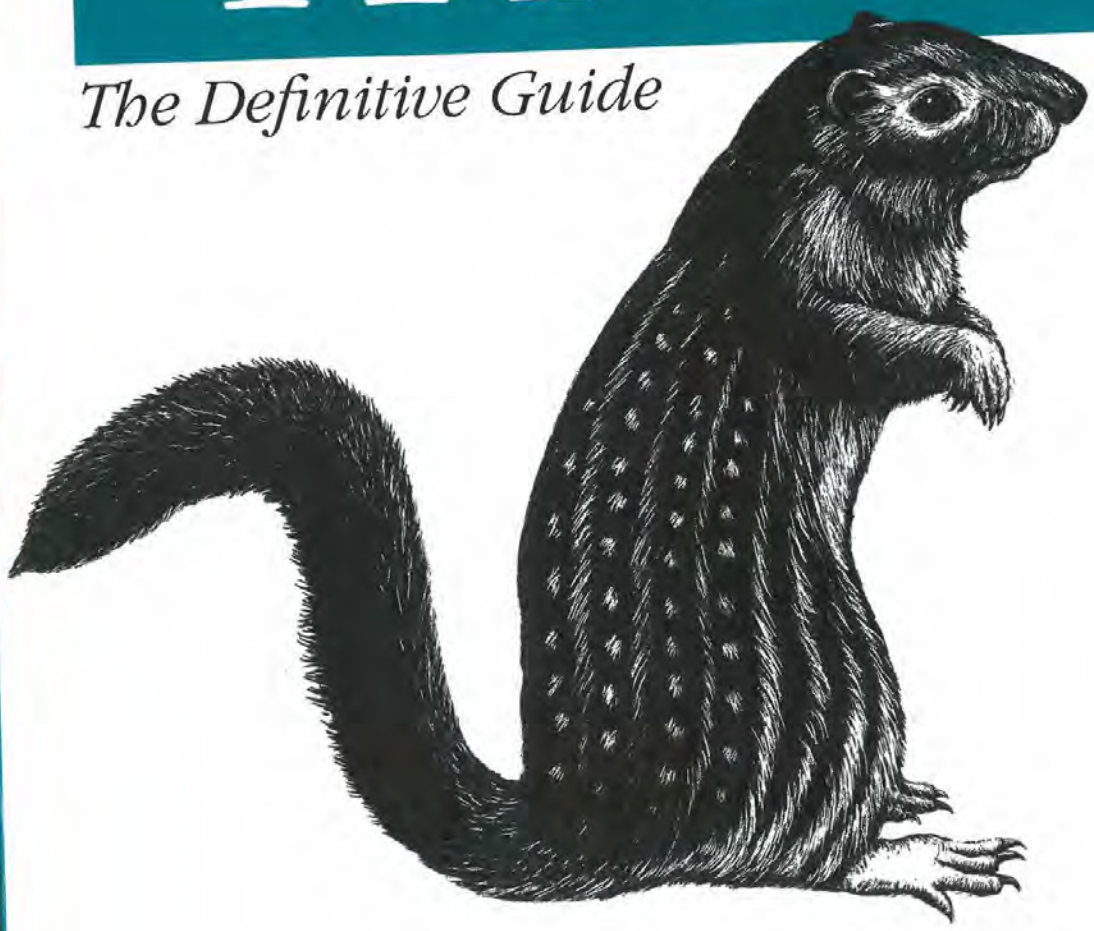
CALL NUMBER	TK6680.5 .W3843 2001 Copy 1
Request in	Jefferson or Adams Building Reading Rooms
Status	Not Charged

Google Exhibit No. 1063
Page 0401

Understanding Web Internals

HTTP

The Definitive Guide



David Gourley & Brian Totty
with Marjorie Sayer, Sailu Reddy & Anshu Aggarwal

O'REILLY®

Google Exhibit No. 1006
Page 001

Google Exhibit No. 1063
Page 0402

HTTP

The Definitive Guide

David Gourley and Brian Totty
with Marjorie Sayer, Sailu Reddy, and Anshu Aggarwal

O'REILLY®
Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

Google Exhibit No. 1006
Page 002

Google Exhibit No. 1063
Page 0403

HTTP: The Definitive Guide
by David Gourley and Brian Totty
with Marjorie Sayer, Sailu Reddy, and Anshu Aggarwal

Copyright © 2002 O'Reilly Media, Inc. All rights reserved.
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media, Inc. books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor:	Linda Mui
Production Editor:	Rachel Wheeler
Cover Designer:	Ellie Volckhausen
Interior Designers:	David Futato and Melanie Wang

Printing History:

September 2002: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *HTTP: The Definitive Guide*, the image of a thirteen-lined ground squirrel, and related trade dress are trademarks of O'Reilly Media, Inc. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-56592-509-0
[LSI] [2011-01-27]

Google Exhibit No. 1006
Page 003

Google Exhibit No. 1063
Page 0404

Table of Contents

Preface xiii

Part I. HTTP: The Web's Foundation

1. Overview of HTTP 3
HTTP: The Internet's Multimedia Courier 3
Web Clients and Servers 4
Resources 4
Transactions 8
Messages 10
Connections 11
Protocol Versions 16
Architectural Components of the Web 17
The End of the Beginning 21
For More Information 21

2. URLs and Resources 23
Navigating the Internet's Resources 24
URL Syntax 26
URL Shortcuts 30
Shady Characters 35
A Sea of Schemes 38
The Future 40
For More Information 41

3. HTTP Messages 43
The Flow of Messages 43
The Parts of a Message 44

Google Exhibit No. 1006
Page 004

Google Exhibit No. 1063
Page 0405

Methods 53
Status Codes 59
Headers 67
For More Information 73

4. Connection Management 74
TCP Connections 74
TCP Performance Considerations 80
HTTP Connection Handling 86
Parallel Connections 88
Persistent Connections 90
Pipelined Connections 99
The Mysteries of Connection Close 101
For More Information 104

Part II. HTTP Architecture

5. Web Servers 109
Web Servers Come in All Shapes and Sizes 109
A Minimal Perl Web Server 111
What Real Web Servers Do 113
Step 1: Accepting Client Connections 115
Step 2: Receiving Request Messages 116
Step 3: Processing Requests 120
Step 4: Mapping and Accessing Resources 120
Step 5: Building Responses 125
Step 6: Sending Responses 127
Step 7: Logging 127
For More Information 127

6. Proxies 129
Web Intermediaries 129
Why Use Proxies? 131
Where Do Proxies Go? 137
Client Proxy Settings 141
Tricky Things About Proxy Requests 144
Tracing Messages 150
Proxy Authentication 156

Google Exhibit No. 1006
Page 005

Google Exhibit No. 1063
Page 0406

Proxy Interoperation 157
For More Information 160

7. Caching 161
Redundant Data Transfers 161
Bandwidth Bottlenecks 161
Flash Crowds 163
Distance Delays 163
Hits and Misses 164
Cache Topologies 168
Cache Processing Steps 171
Keeping Copies Fresh 175
Controlling Cachability 182
Setting Cache Controls 186
Detailed Algorithms 187
Caches and Advertising 194
For More Information 196

8. Integration Points: Gateways, Tunnels, and Relays 197
Gateways 197
Protocol Gateways 200
Resource Gateways 203
Application Interfaces and Web Services 205
Tunnels 206
Relays 212
For More Information 213

9. Web Robots 215
Crawlers and Crawling 215
Robotic HTTP 225
Misbehaving Robots 228
Excluding Robots 229
Robot Etiquette 239
Search Engines 242
For More Information 246

10. HTTP-NG 247
HTTP's Growing Pains 247
HTTP-NG Activity 248

Google Exhibit No. 1006
Page 006

Google Exhibit No. 1063
Page 0407

Modularize and Enhance 248
Distributed Objects 249
Layer 1: Messaging 250
Layer 2: Remote Invocation 250
Layer 3: Web Application 251
WebMUX 251
Binary Wire Protocol 252
Current Status 252
For More Information 253

Part III. Identification, Authorization, and Security

11. Client Identification and Cookies 257
The Personal Touch 257
HTTP Headers 258
Client IP Address 259
User Login 260
Fat URLs 262
Cookies 263
For More Information 276

12. Basic Authentication 277
Authentication 277
Basic Authentication 281
The Security Flaws of Basic Authentication 283
For More Information 285

13. Digest Authentication 286
The Improvements of Digest Authentication 286
Digest Calculations 291
Quality of Protection Enhancements 299
Practical Considerations 300
Security Considerations 303
For More Information 306

14. Secure HTTP 307
Making HTTP Safe 307
Digital Cryptography 309

Google Exhibit No. 1006
Page 007

Google Exhibit No. 1063
Page 0408

Symmetric-Key Cryptography 313
Public-Key Cryptography 315
Digital Signatures 317
Digital Certificates 319
HTTPS: The Details 322
A Real HTTPS Client 328
Tunneling Secure Traffic Through Proxies 335
For More Information 336

Part IV. Entities, Encodings, and Internationalization

15. Entities and Encodings 341
Messages Are Crates, Entities Are Cargo 342
Content-Length: The Entity's Size 344
Entity Digests 347
Media Type and Charset 348
Content Encoding 351
Transfer Encoding and Chunked Encoding 354
Time-Varying Instances 359
Validators and Freshness 360
Range Requests 363
Delta Encoding 365
For More Information 369

16. Internationalization 370
HTTP Support for International Content 370
Character Sets and HTTP 371
Multilingual Character Encoding Primer 376
Language Tags and HTTP 384
Internationalized URIs 389
Other Considerations 392
For More Information 392

17. Content Negotiation and Transcoding 395
Content-Negotiation Techniques 395
Client-Driven Negotiation 396
Server-Driven Negotiation 397
Transparent Negotiation 400

Google Exhibit No. 1006
Page 008

Google Exhibit No. 1063
Page 0409

Transcoding 403
Next Steps 405
For More Information 406

Part V. Content Publishing and Distribution

18. Web Hosting 411
Hosting Services 411
Virtual Hosting 413
Making Web Sites Reliable 419
Making Web Sites Fast 422
For More Information 423

19. Publishing Systems 424
FrontPage Server Extensions for Publishing Support 424
WebDAV and Collaborative Authoring 429
For More Information 446

20. Redirection and Load Balancing 448
Why Redirect? 449
Where to Redirect 449
Overview of Redirection Protocols 450
General Redirection Methods 452
Proxy Redirection Methods 462
Cache Redirection Methods 469
Internet Cache Protocol 473
Cache Array Routing Protocol 475
Hyper Text Caching Protocol 478
For More Information 481

21. Logging and Usage Tracking 483
What to Log? 483
Log Formats 484
Hit Metering 492
A Word on Privacy 495
For More Information 495

Google Exhibit No. 1006
Page 009

Google Exhibit No. 1063
Page 0410

Part VI. Appendixes

A. URI Schemes 499

B. HTTP Status Codes 505

C. HTTP Header Reference 508

D. MIME Types 533

E. Base-64 Encoding 570

F. Digest Authentication 574

G. Language Tags 581

H. MIME Charset Registry 602

Index 617

Google Exhibit No. 1006
Page 010

Google Exhibit No. 1063
Page 0411

CHAPTER 1

Overview of HTTP

The world's web browsers, servers, and related web applications all talk to each other through HTTP, the Hypertext Transfer Protocol. HTTP is the common language of the modern global Internet.

This chapter is a concise overview of HTTP. You'll see how web applications use HTTP to communicate, and you'll get a rough idea of how HTTP does its job. In particular, we talk about:

- How web clients and servers communicate
- Where resources (web content) come from
- How web transactions work
- The format of the messages used for HTTP communication
- The underlying TCP network transport
- The different variations of the HTTP protocol
- Some of the many HTTP architectural components installed around the Internet

We've got a lot of ground to cover, so let's get started on our tour of HTTP.

HTTP: The Internet's Multimedia Courier

Billions of JPEG images, HTML pages, text files, MPEG movies, WAV audio files, Java applets, and more cruise through the Internet each and every day. HTTP moves the bulk of this information quickly, conveniently, and reliably from web servers all around the world to web browsers on people's desktops.

Because HTTP uses reliable data-transmission protocols, it guarantees that your data will not be damaged or scrambled in transit, even when it comes from the other side of the globe. This is good for you as a user, because you can access information without worrying about its integrity. Reliable transmission is also good for you as an Internet application developer, because you don't have to worry about HTTP communications

Google Exhibit No. 1006
Page 011

Google Exhibit No. 1063
Page 0412

being destroyed, duplicated, or distorted in transit. You can focus on programming the distinguishing details of your application, without worrying about the flaws and foibles of the Internet.

Let's look more closely at how HTTP transports the Web's traffic.

Web Clients and Servers

Web content lives on web servers. Web servers speak the HTTP protocol, so they are often called HTTP servers. These HTTP servers store the Internet's data and provide the data when it is requested by HTTP clients. The clients send HTTP requests to servers, and servers return the requested data in HTTP responses, as sketched in Figure 1-1. Together, HTTP clients and HTTP servers make up the basic components of the World Wide Web.

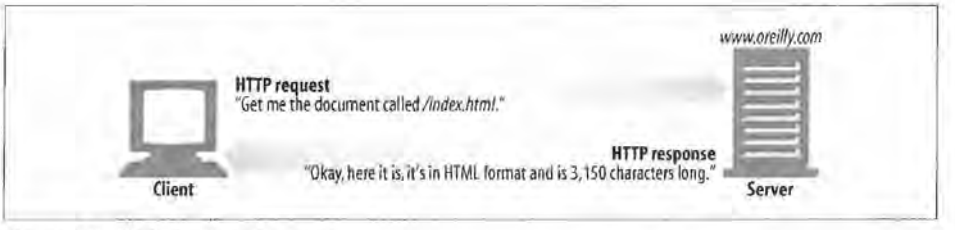


Figure 1-1. Web clients and servers

You probably use HTTP clients every day. The most common client is a web browser, such as Microsoft Internet Explorer or Netscape Navigator. Web browsers request HTTP objects from servers and display the objects on your screen.

When you browse to a page, such as "http://www.oreilly.com/index.html," your browser sends an HTTP request to the server *www.oreilly.com* (see Figure 1-1). The server tries to find the desired object (in this case, "/index.html") and, if successful, sends the object to the client in an HTTP response, along with the type of the object, the length of the object, and other information.

Resources

Web servers host *web resources*. A web resource is the source of web content. The simplest kind of web resource is a static file on the web server's filesystem. These files can contain anything: they might be text files, HTML files, Microsoft Word files, Adobe Acrobat files, JPEG image files, AVI movie files, or any other format you can think of.

However, resources don't have to be static files. Resources can also be software programs that generate content on demand. These dynamic content resources can generate content based on your identity, on what information you've requested, or on

Google Exhibit No. 1006
Page 012

Google Exhibit No. 1063
Page 0413

the time of day. They can show you a live image from a camera, or let you trade stocks, search real estate databases, or buy gifts from online stores (see Figure 1-2).

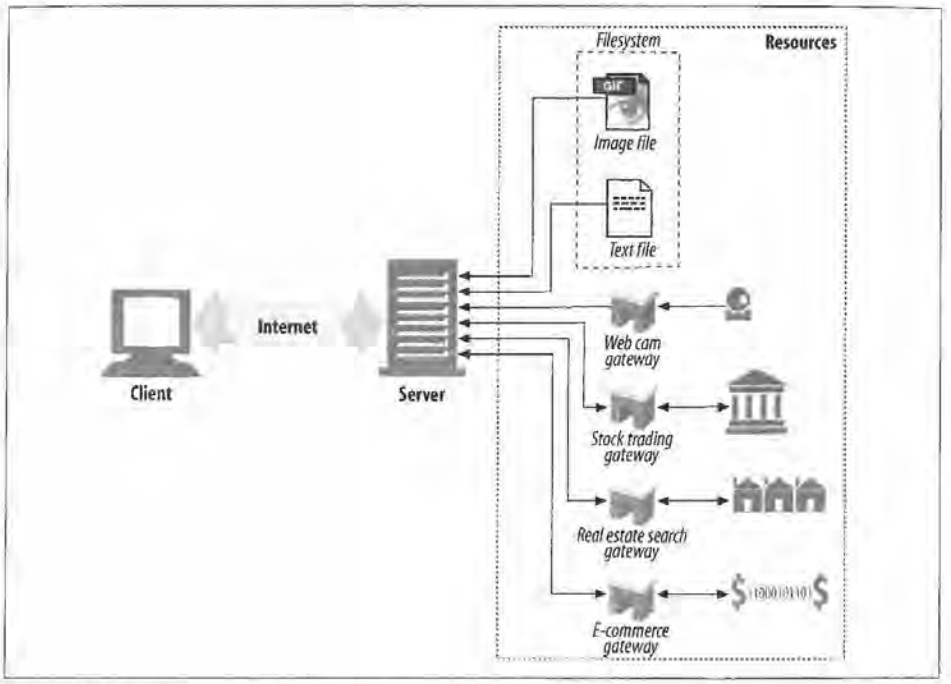


Figure 1-2. A web resource is anything that provides web content

In summary, a resource is any kind of content source. A file containing your company's sales forecast spreadsheet is a resource. A web gateway to scan your local public library's shelves is a resource. An Internet search engine is a resource.

Media Types

Because the Internet hosts many thousands of different data types, HTTP carefully tags each object being transported through the Web with a data format label called a *MIME type*. MIME (Multipurpose Internet Mail Extensions) was originally designed to solve problems encountered in moving messages between different electronic mail systems. MIME worked so well for email that HTTP adopted it to describe and label its own multimedia content.

Web servers attach a MIME type to all HTTP object data (see Figure 1-3). When a web browser gets an object back from a server, it looks at the associated MIME type to see if it knows how to handle the object. Most browsers can handle hundreds of popular object types: displaying image files, parsing and formatting HTML files, playing audio files through the computer's speakers, or launching external plug-in software to handle special formats.

Google Exhibit No. 1006
Page 013

Google Exhibit No. 1063
Page 0414

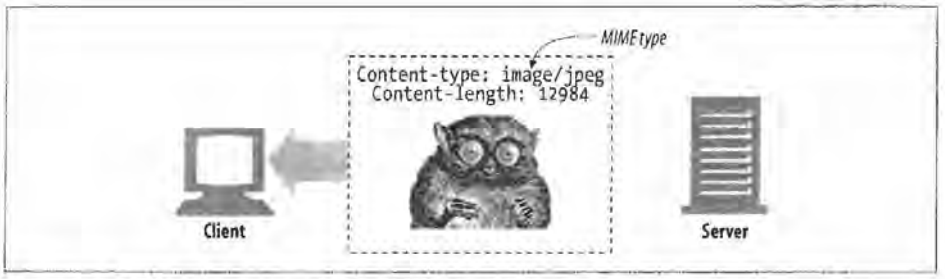


Figure 1-3, MIME types are sent back with the data content

A MIME type is a textual label, represented as a primary object type and a specific subtype, separated by a slash. For example:

- An HTML-formatted text document would be labeled with type `text/html`.
- A plain ASCII text document would be labeled with type `text/plain`.
- A JPEG version of an image would be `image/jpeg`.
- A GIF-format image would be `image/gif`.
- An Apple QuickTime movie would be `video/quicktime`.
- A Microsoft PowerPoint presentation would be `application/vnd.ms-powerpoint`.

There are hundreds of popular MIME types, and many more experimental or limited-use types. A very thorough MIME type list is provided in Appendix D.

URIs

Each web server resource has a name, so clients can point out what resources they are interested in. The server resource name is called a *uniform resource identifier*, or URI. URIs are like the postal addresses of the Internet, uniquely identifying and locating information resources around the world.

Here's a URI for an image resource on Joe's Hardware store's web server:

```
http://www.joes-hardware.com/specials/saw-blade.gif
```

Figure 1-4 shows how the URI specifies the HTTP protocol to access the saw-blade GIF resource on Joe's store's server. Given the URI, HTTP can retrieve the object. URIs come in two flavors, called URLs and URNs. Let's take a peek at each of these types of resource identifiers now.

URLs

The *uniform resource locator* (URL) is the most common form of resource identifier. URLs describe the specific location of a resource on a particular server. They tell you exactly how to fetch a resource from a precise, fixed location. Figure 1-4 shows how a URL tells precisely where a resource is located and how to access it. Table 1-1 shows a few examples of URLs.

Google Exhibit No. 1006
Page 014

Google Exhibit No. 1063
Page 0415

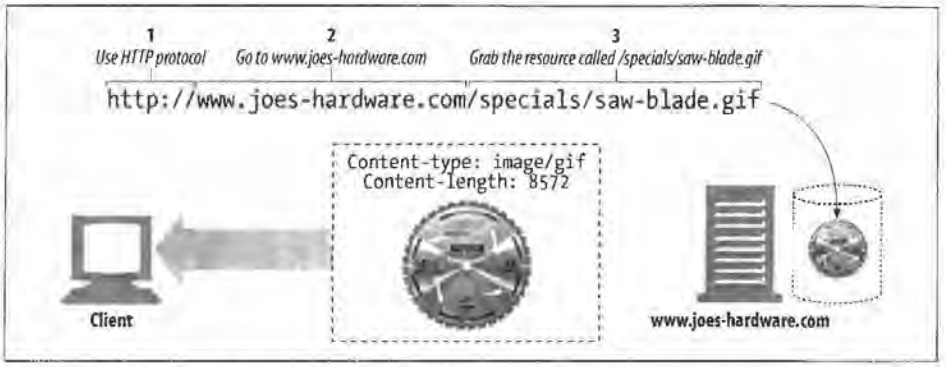


Figure 1-4. URLs specify protocol, server, and local resource

Table 1-1. Example URLs

URL	Description
http://www.oreilly.com/index.html	The home URL for O'Reilly & Associates, Inc.
http://www.yahoo.com/images/logo.gif	The URL for the Yahoo! web site's logo
http://www.joes-hardware.com/inventory-check.cgi?item=12731	The URL for a program that checks if inventory item #12731 is in stock
ftp://joe:tools4u@ftp.joes-hardware.com/locking-pliers.gif	The URL for the *locking-pliers.gif* image file, using password-protected FTP as the access protocol

Most URLs follow a standardized format of three main parts:

- The first part of the URL is called the *scheme*, and it describes the protocol used to access the resource. This is usually the HTTP protocol (*http://*).
- The second part gives the server Internet address (e.g., *www.joes-hardware.com*).
- The rest names a resource on the web server (e.g., */specials/saw-blade.gif*).

Today, almost every URI is a URL.

URNs

The second flavor of URI is the *uniform resource name*, or URN. A URN serves as a unique name for a particular piece of content, independent of where the resource currently resides. These location-independent URNs allow resources to move from place to place. URNs also allow resources to be accessed by multiple network access protocols while maintaining the same name.

For example, the following URN might be used to name the Internet standards document "RFC 2141" regardless of where it resides (it may even be copied in several places):

urn:ietf:rfc:2141

Google Exhibit No. 1006
Page 015

Google Exhibit No. 1063
Page 0416

URNs are still experimental and not yet widely adopted. To work effectively, URNs need a supporting infrastructure to resolve resource locations; the lack of such an infrastructure has also slowed their adoption. But URNs do hold some exciting promise for the future. We'll discuss URNs in a bit more detail in Chapter 2, but most of the remainder of this book focuses almost exclusively on URLs.

Unless stated otherwise, we adopt the conventional terminology and use URI and URL interchangeably for the remainder of this book.

Transactions

Let's look in more detail how clients use HTTP to transact with web servers and their resources. An HTTP transaction consists of a request command (sent from client to server), and a response result (sent from the server back to the client). This communication happens with formatted blocks of data called *HTTP messages*, as illustrated in Figure 1-5.

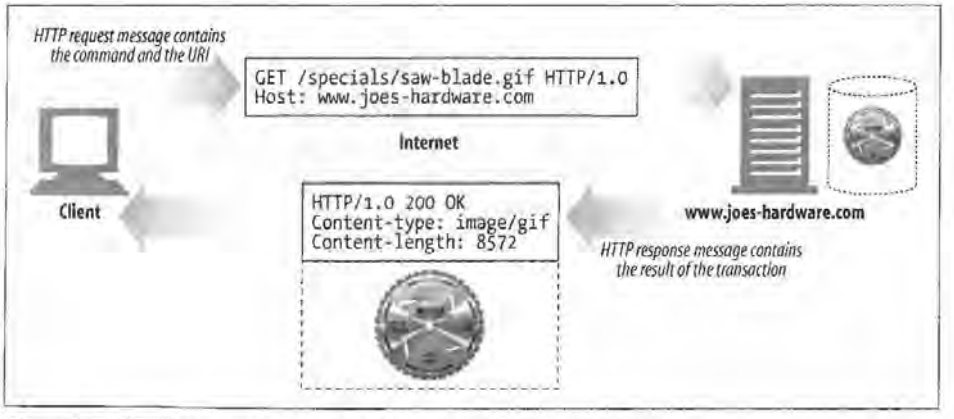


Figure 1-5. HTTP transactions consist of request and response messages

Methods

HTTP supports several different request commands, called *HTTP methods*. Every HTTP request message has a method. The method tells the server what action to perform (fetch a web page, run a gateway program, delete a file, etc.). Table 1-2 lists five common HTTP methods.

Table 1-2. Some common HTTP methods

HTTP method	Description
GET	Send named resource from the server to the client.
PUT	Store data from client into a named server resource.

Google Exhibit No. 1006
Page 016

Google Exhibit No. 1063
Page 0417

Table 1-2. Some common HTTP methods (continued)

HTTP method	Description
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

We'll discuss HTTP methods in detail in Chapter 3.

Status Codes

Every HTTP response message comes back with a status code. The status code is a three-digit numeric code that tells the client if the request succeeded, or if other actions are required. A few common status codes are shown in Table 1-3.

Table 1-3. Some common HTTP status codes

HTTP status code	Description
200	OK. Document returned correctly.
302	Redirect. Go someplace else to get the resource.
404	Not Found. Can't find this resource.

HTTP also sends an explanatory textual "reason phrase" with each numeric status code (see the response message in Figure 1-5). The textual phrase is included only for descriptive purposes; the numeric code is used for all processing.

The following status codes and reason phrases are treated identically by HTTP software:

```
200 OK
200 Document attached
200 Success
200 All's cool, dude
```

HTTP status codes are explained in detail in Chapter 3.

Web Pages Can Consist of Multiple Objects

An application often issues multiple HTTP transactions to accomplish a task. For example, a web browser issues a cascade of HTTP transactions to fetch and display a graphics-rich web page. The browser performs one transaction to fetch the HTML "skeleton" that describes the page layout, then issues additional HTTP transactions for each embedded image, graphics pane, Java applet, etc. These embedded resources might even reside on different servers, as shown in Figure 1-6. Thus, a "web page" often is a collection of resources, not a single resource.

Google Exhibit No. 1006
Page 017

Google Exhibit No. 1063
Page 0418

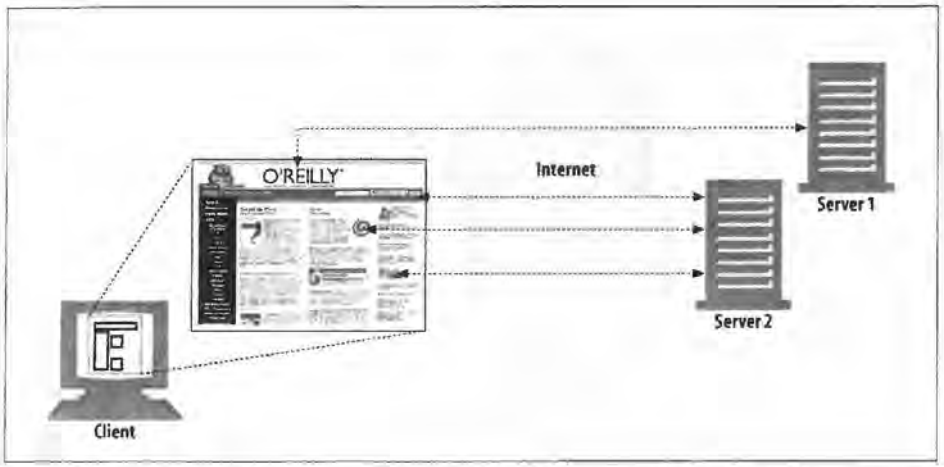


Figure 1-6. Composite web pages require separate HTTP transactions for each embedded resource

Messages

Now let's take a quick look at the structure of HTTP request and response messages. We'll study HTTP messages in exquisite detail in Chapter 3.

HTTP messages are simple, line-oriented sequences of characters. Because they are plain text, not binary, they are easy for humans to read and write.* Figure 1-7 shows the HTTP messages for a simple transaction.

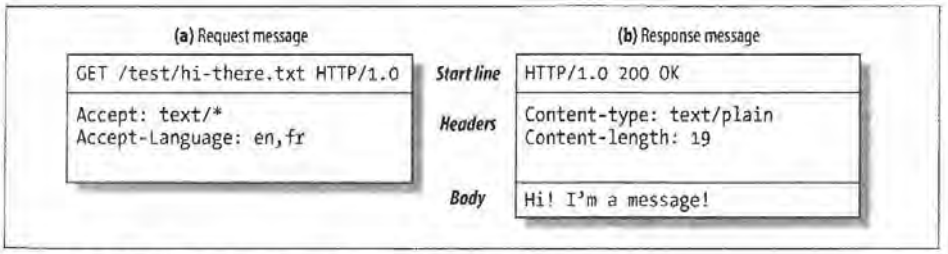


Figure 1-7. HTTP messages have a simple, line-oriented text structure

HTTP messages sent from web clients to web servers are called *request messages*. Messages from servers to clients are called *response messages*. There are no other kinds of HTTP messages. The formats of HTTP request and response messages are very similar.

* Some programmers complain about the difficulty of HTTP parsing, which can be tricky and error-prone, especially when designing high-speed software. A binary format or a more restricted text format might have been simpler to process, but most HTTP programmers appreciate HTTP's extensibility and debuggability.

Google Exhibit No. 1006
Page 018

Google Exhibit No. 1063
Page 0419

HTTP messages consist of three parts:

Start line
: The first line of the message is the start line, indicating what to do for a request or what happened for a response.

Header fields
: Zero or more header fields follow the start line. Each header field consists of a name and a value, separated by a colon (:) for easy parsing. The headers end with a blank line. Adding a header field is as easy as adding another line.

Body
: After the blank line is an optional message body containing any kind of data. Request bodies carry data to the web server; response bodies carry data back to the client. Unlike the start lines and headers, which are textual and structured, the body can contain arbitrary binary data (e.g., images, videos, audio tracks, software applications). Of course, the body can also contain text.

Simple Message Example

Figure 1-8 shows the HTTP messages that might be sent as part of a simple transaction. The browser requests the resource `http://www.joes-hardware.com/tools.html`.

In Figure 1-8, the browser sends an HTTP request message. The request has a GET method in the start line, and the local resource is */tools.html*. The request indicates it is speaking Version 1.0 of the HTTP protocol. The request message has no body, because no request data is needed to GET a simple document from a server.

The server sends back an HTTP response message. The response contains the HTTP version number (HTTP/1.0), a success status code (200), a descriptive reason phrase (OK), and a block of response header fields, all followed by the response body containing the requested document. The response body length is noted in the Content-Length header, and the document's MIME type is noted in the Content-Type header.

Connections

Now that we've sketched what HTTP's messages look like, let's talk for a moment about how messages move from place to place, across Transmission Control Protocol (TCP) connections.

TCP/IP

HTTP is an application layer protocol. HTTP doesn't worry about the nitty-gritty details of network communication; instead, it leaves the details of networking to TCP/IP, the popular reliable Internet transport protocol.

Google Exhibit No. 1006
Page 019

Google Exhibit No. 1063
Page 0420

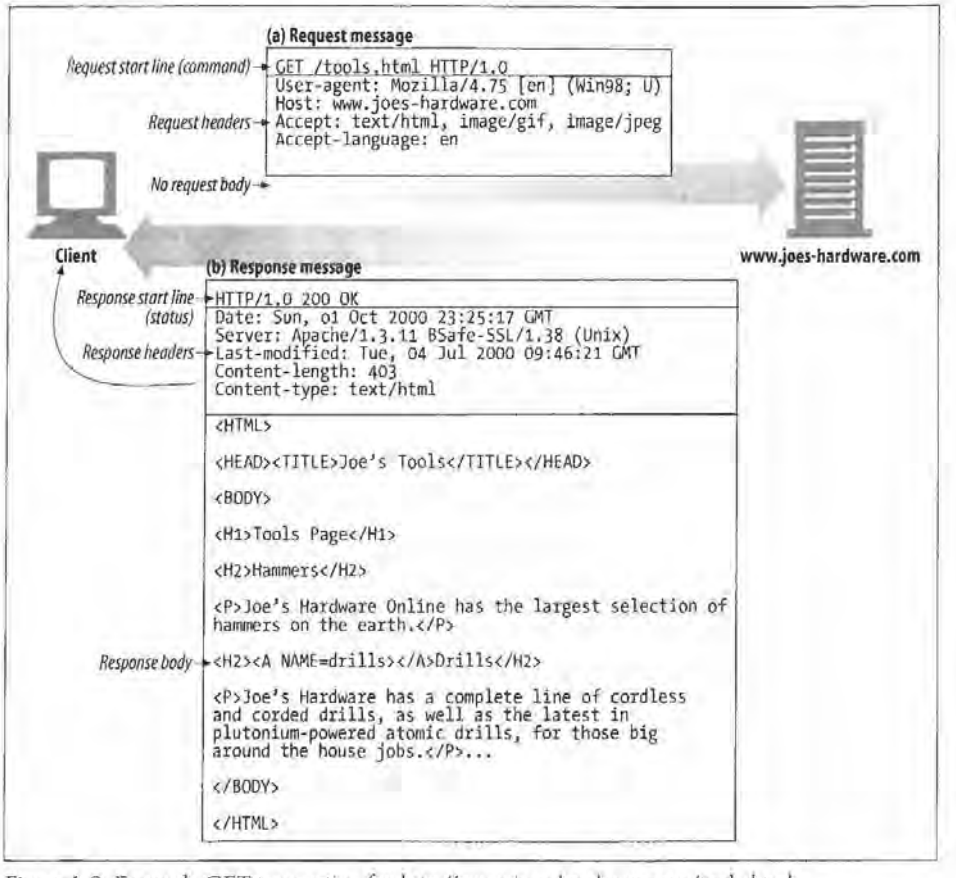


Figure 1-8. Example GET transaction for http://www.joes-hardware.com/tools.html

TCP provides:

- Error-free data transportation
- In-order delivery (data will always arrive in the order in which it was sent)
- Unsegmented data stream (can dribble out data in any size at any time)

The Internet itself is based on TCP/IP, a popular layered set of packet-switched network protocols spoken by computers and network devices around the world. TCP/IP hides the peculiarities and foibles of individual networks and hardware, letting computers and networks of any type talk together reliably.

Once a TCP connection is established, messages exchanged between the client and server computers will never be lost, damaged, or received out of order.

In networking terms, the HTTP protocol is *layered* over TCP. HTTP uses TCP to transport its message data. Likewise, TCP is layered over IP (see Figure 1-9).

Google Exhibit No. 1006
Page 020

Google Exhibit No. 1063
Page 0421

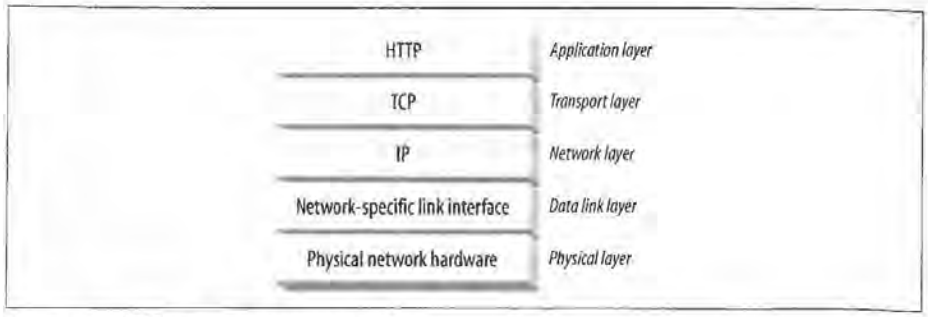


Figure 1-9. HTTP network protocol stack

Connections, IP Addresses, and Port Numbers

Before an HTTP client can send a message to a server, it needs to establish a TCP/IP connection between the client and server using Internet protocol (IP) addresses and port numbers.

Setting up a TCP connection is sort of like calling someone at a corporate office. First, you dial the company's phone number. This gets you to the right organization. Then, you dial the specific extension of the person you're trying to reach.

In TCP, you need the IP address of the server computer and the TCP port number associated with the specific software program running on the server.

This is all well and good, but how do you get the IP address and port number of the HTTP server in the first place? Why, the URL, of course! We mentioned before that URLs are the addresses for resources, so naturally enough they can provide us with the IP address for the machine that has the resource. Let's take a look at a few URLs:

```
http://207.200.83.29:80/index.html
http://www.netscape.com:80/index.html
http://www.netscape.com/index.html
```

The first URL has the machine's IP address, "207.200.83.29", and port number, "80".

The second URL doesn't have a numeric IP address; it has a textual domain name, or *hostname* ("www.netscape.com"). The hostname is just a human-friendly alias for an IP address. Hostnames can easily be converted into IP addresses through a facility called the Domain Name Service (DNS), so we're all set here, too. We will talk much more about DNS and URLs in Chapter 2.

The final URL has no port number. When the port number is missing from an HTTP URL, you can assume the default value of port 80.

With the IP address and port number, a client can easily communicate via TCP/IP. Figure 1-10 shows how a browser uses HTTP to display a simple HTML resource that resides on a distant server.

Google Exhibit No. 1006
Page 021

Google Exhibit No. 1063
Page 0422

Here are the steps:

(a) The browser extracts the server's hostname from the URL.
(b) The browser converts the server's hostname into the server's IP address.
(c) The browser extracts the port number (if any) from the URL.
(d) The browser establishes a TCP connection with the web server.
(e) The browser sends an HTTP request message to the server.
(f) The server sends an HTTP response back to the browser.
(g) The connection is closed, and the browser displays the document.

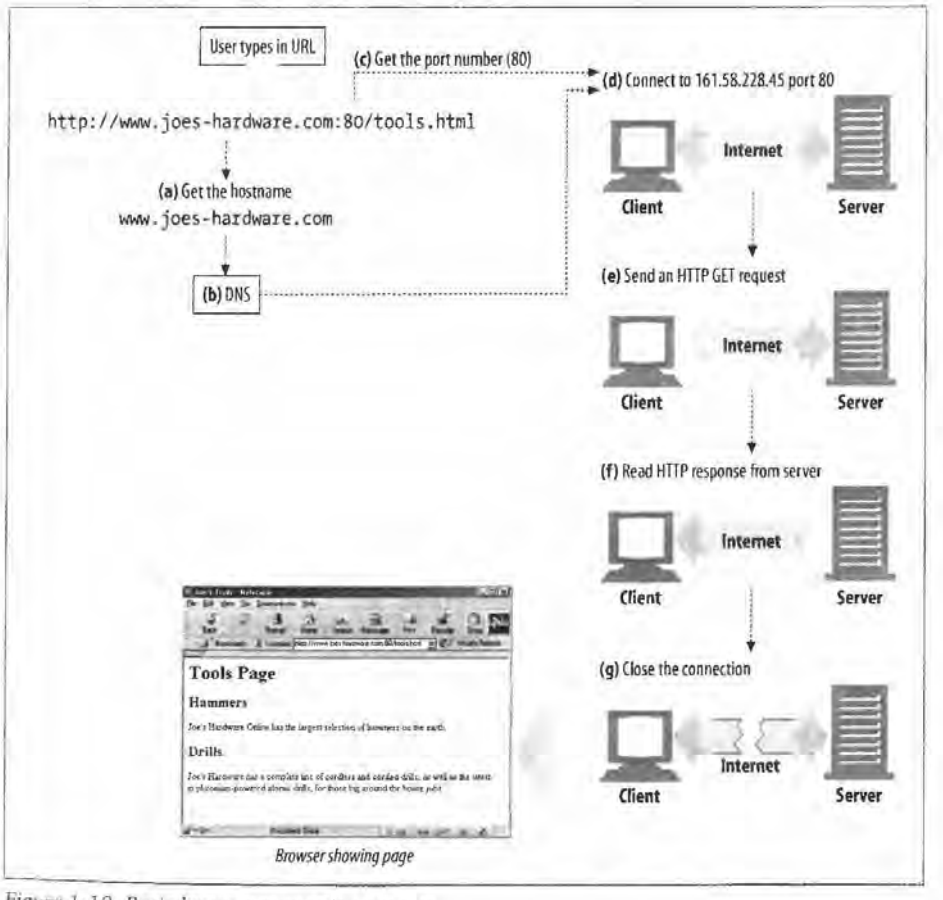


Figure 1-10. Basic browser connection process

Google Exhibit No. 1006
Page 022

Google Exhibit No. 1063
Page 0423

A Real Example Using Telnet

Because HTTP uses TCP/IP, and is text-based, as opposed to using some obscure binary format, it is simple to talk directly to a web server.

The Telnet utility connects your keyboard to a destination TCP port and connects the TCP port output back to your display screen. Telnet is commonly used for remote terminal sessions, but it can generally connect to any TCP server, including HTTP servers.

You can use the Telnet utility to talk directly to web servers. Telnet lets you open a TCP connection to a port on a machine and type characters directly into the port. The web server treats you as a web client, and any data sent back on the TCP connection is displayed onscreen.

Let's use Telnet to interact with a real web server. We will use Telnet to fetch the document pointed to by the URL *http://www.joes-hardware.com:80/tools.html* (you can try this example yourself).

Let's review what should happen:

- First, we need to look up the IP address of *www.joes-hardware.com* and open a TCP connection to port 80 on that machine. Telnet does this legwork for us.
- Once the TCP connection is open, we need to type in the HTTP request.
- When the request is complete (indicated by a blank line), the server should send back the content in an HTTP response and close the connection.

Our example HTTP request for *http://www.joes-hardware.com:80/tools.html* is shown in Example 1-1. What we typed is shown in boldface.

Example 1-1. An HTTP transaction using telnet

```
% telnet www.joes-hardware.com 80
Trying 161.58.228.45...
Connected to joes-hardware.com.
Escape character is '^]'.
GET /tools.html HTTP/1.1
Host: www.joes-hardware.com

HTTP/1.1 200 OK
Date: Sun, 01 Oct 2000 23:25:17 GMT
Server: Apache/1.3.11 BSafe-SSL/1.38 (Unix) FrontPage/4.0.4.3
Last-Modified: Tue, 04 Jul 2000 09:46:21 GMT
ETag: "373979-193-3961b26d"
Accept-Ranges: bytes
Content-Length: 403
Connection: close
Content-Type: text/html
```


Google Exhibit No. 1006
Page 023

Google Exhibit No. 1063
Page 0424

Example 1-1. An HTTP transaction using telnet (continued)

```
<HTML>
<HEAD><TITLE>Joe's Tools</TITLE></HEAD>
<BODY>
<H1>Tools Page</H1>
<H2>Hammers</H2>
<P>Joe's Hardware Online has the largest selection of hammers on the earth.</P>
<H2><A NAME=drills></A>Drills</H2>
<P>Joe's Hardware has a complete line of cordless and corded drills, as well as the latest
in plutonium-powered atomic drills, for those big around the house jobs.</P> ...
</BODY>
</HTML>
Connection closed by foreign host.
```

Telnet looks up the hostname and opens a connection to the *www.joes-hardware.com* web server, which is listening on port 80. The three lines after the command are output from Telnet, telling us it has established a connection.

We then type in our basic request command, "GET /tools.html HTTP/1.1", and send a Host header providing the original hostname, followed by a blank line, asking the server to GET us the resource "/tools.html" from the server *www.joes-hardware.com*. After that, the server responds with a response line, several response headers, a blank line, and finally the body of the HTML document.

Beware that Telnet mimics HTTP clients well but doesn't work well as a server. And automated Telnet scripting is no fun at all. For a more flexible tool, you might want to check out nc (netcat). The nc tool lets you easily manipulate and script UDP- and TCP-based traffic, including HTTP. See *http://netcat.sourceforge.net* for details.

Protocol Versions

There are several versions of the HTTP protocol in use today. HTTP applications need to work hard to robustly handle different variations of the HTTP protocol. The versions in use are:

HTTP/0.9

The 1991 prototype version of HTTP is known as HTTP/0.9. This protocol contains many serious design flaws and should be used only to interoperate with legacy clients. HTTP/0.9 supports only the GET method, and it does not support MIME typing of multimedia content, HTTP headers, or version numbers. HTTP/0.9 was originally defined to fetch simple HTML objects. It was soon replaced with HTTP/1.0.

HTTP/1.0

1.0 was the first version of HTTP that was widely deployed. HTTP/1.0 added version numbers, HTTP headers, additional methods, and multimedia object handling. HTTP/1.0 made it practical to support graphically appealing web

Google Exhibit No. 1006
Page 024

Google Exhibit No. 1063
Page 0425

pages and interactive forms, which helped promote the wide-scale adoption of the World Wide Web. This specification was never well specified. It represented a collection of best practices in a time of rapid commercial and academic evolution of the protocol.

HTTP/1.0+
: Many popular web clients and servers rapidly added features to HTTP in the mid-1990s to meet the demands of a rapidly expanding, commercially successful World Wide Web. Many of these features, including long-lasting "keep-alive" connections, virtual hosting support, and proxy connection support, were added to HTTP and became unofficial, de facto standards. This informal, extended version of HTTP is often referred to as HTTP/1.0+.

HTTP/1.1
: HTTP/1.1 focused on correcting architectural flaws in the design of HTTP, specifying semantics, introducing significant performance optimizations, and removing mis-features. HTTP/1.1 also included support for the more sophisticated web applications and deployments that were under way in the late 1990s. HTTP/1.1 is the current version of HTTP.

HTTP-NG (a.k.a. HTTP/2.0)
: HTTP-NG is a prototype proposal for an architectural successor to HTTP/1.1 that focuses on significant performance optimizations and a more powerful framework for remote execution of server logic. The HTTP-NG research effort concluded in 1998, and at the time of this writing, there are no plans to advance this proposal as a replacement for HTTP/1.1. See Chapter 10 for more information.

Architectural Components of the Web

In this overview chapter, we've focused on how two web applications (web browsers and web servers) send messages back and forth to implement basic transactions. There are many other web applications that you interact with on the Internet. In this section, we'll outline several other important applications, including:

Proxies
: HTTP intermediaries that sit between clients and servers

Caches
: HTTP storehouses that keep copies of popular web pages close to clients

Gateways
: Special web servers that connect to other applications

Tunnels
: Special proxies that blindly forward HTTP communications

Agents
: Semi-intelligent web clients that make automated HTTP requests

Google Exhibit No. 1006
Page 025

Google Exhibit No. 1063
Page 0426

Proxies

Let's start by looking at HTTP *proxy servers*, important building blocks for web security, application integration, and performance optimization.

As shown in Figure 1-11, a proxy sits between a client and a server, receiving all of the client's HTTP requests and relaying the requests to the server (perhaps after modifying the requests). These applications act as a proxy for the user, accessing the server on the user's behalf.

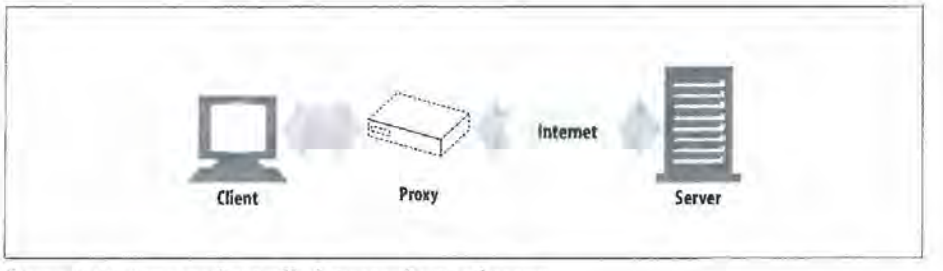


Figure 1-11. Proxies relay traffic between client and server

Proxies are often used for security, acting as trusted intermediaries through which all web traffic flows. Proxies can also filter requests and responses; for example, to detect application viruses in corporate downloads or to filter adult content away from elementary-school students. We'll talk about proxies in detail in Chapter 6.

Caches

A *web cache* or *caching proxy* is a special type of HTTP proxy server that keeps copies of popular documents that pass through the proxy. The next client requesting the same document can be served from the cache's personal copy (see Figure 1-12).

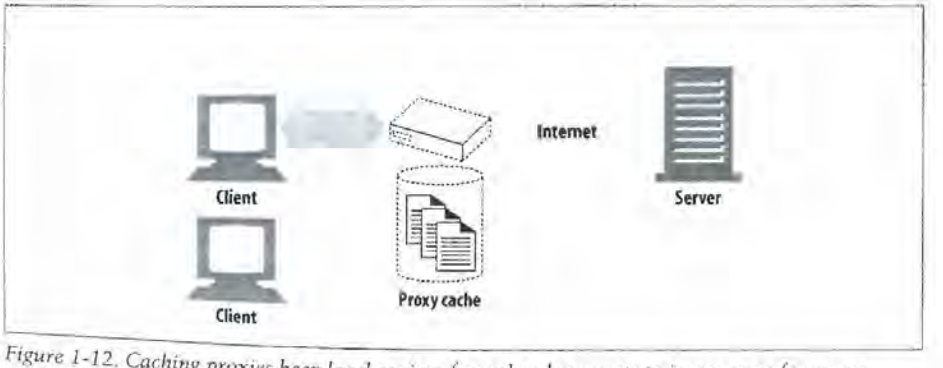


Figure 1-12. Caching proxies keep local copies of popular documents to improve performance

Google Exhibit No. 1006
Page 026

Google Exhibit No. 1063
Page 0427

A client may be able to download a document much more quickly from a nearby cache than from a distant web server. HTTP defines many facilities to make caching more effective and to regulate the freshness and privacy of cached content. We cover caching technology in Chapter 7.

Gateways

Gateways are special servers that act as intermediaries for other servers. They are often used to convert HTTP traffic to another protocol. A gateway always receives requests as if it was the origin server for the resource. The client may not be aware it is communicating with a gateway.

For example, an HTTP/FTP gateway receives requests for FTP URIs via HTTP requests but fetches the documents using the FTP protocol (see Figure 1-13). The resulting document is packed into an HTTP message and sent to the client. We discuss gateways in Chapter 8.

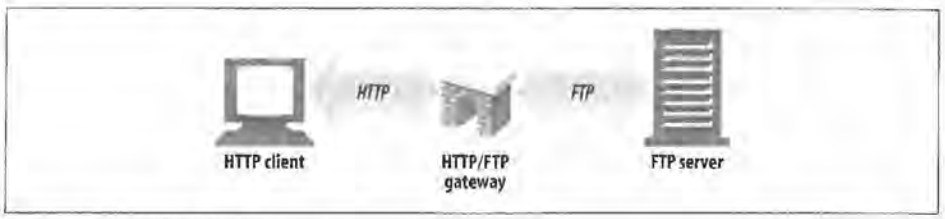


Figure 1-13. HTTP/FTP gateway

Tunnels

Tunnels are HTTP applications that, after setup, blindly relay raw data between two connections. HTTP tunnels are often used to transport non-HTTP data over one or more HTTP connections, without looking at the data.

One popular use of HTTP tunnels is to carry encrypted Secure Sockets Layer (SSL) traffic through an HTTP connection, allowing SSL traffic through corporate firewalls that permit only web traffic. As sketched in Figure 1-14, an HTTP/SSL tunnel receives an HTTP request to establish an outgoing connection to a destination address and port, then proceeds to tunnel the encrypted SSL traffic over the HTTP channel so that it can be blindly relayed to the destination server.

Agents

User agents (or just *agents*) are client programs that make HTTP requests on the user's behalf. Any application that issues web requests is an HTTP agent. So far, we've talked about only one kind of HTTP agent: web browsers. But there are many other kinds of user agents.

Google Exhibit No. 1006
Page 027

Google Exhibit No. 1063
Page 0428

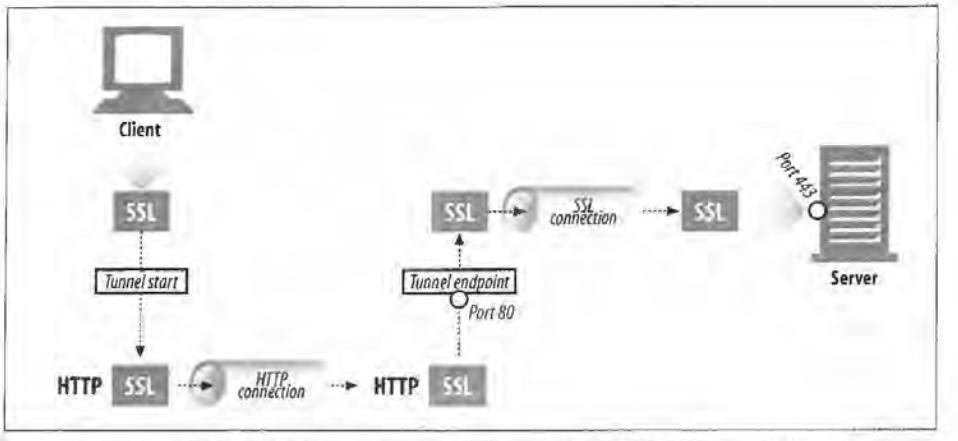


Figure 1-14. Tunnels forward data across non-HTTP networks (HTTP/SSL tunnel shown)

For example, there are machine-automated user agents that autonomously wander the Web, issuing HTTP transactions and fetching content, without human supervision. These automated agents often have colorful names, such as "spiders" or "web robots" (see Figure 1-15). Spiders wander the Web to build useful archives of web content, such as a search engine's database or a product catalog for a comparison-shopping robot. See Chapter 9 for more information.

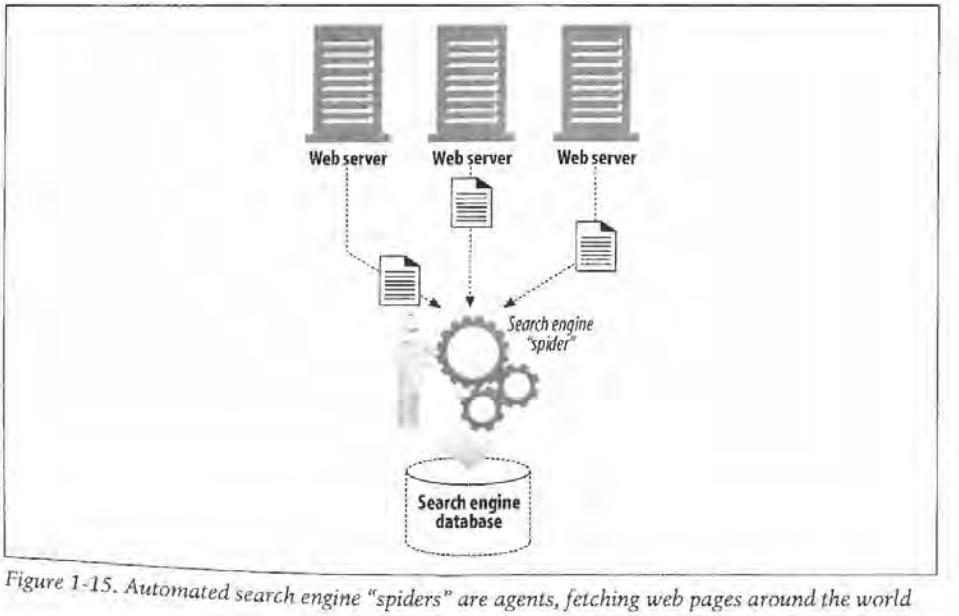


Figure 1-15. Automated search engine "spiders" are agents, fetching web pages around the world

Google Exhibit No. 1006
Page 028

Google Exhibit No. 1063
Page 0429

The End of the Beginning

That's it for our quick introduction to HTTP. In this chapter, we highlighted HTTP's role as a multimedia transport protocol. We outlined how HTTP uses URIs to name multimedia resources on remote servers, we sketched how HTTP request and response messages are used to manipulate multimedia resources on remote servers, and we finished by surveying a few of the web applications that use HTTP.

The remaining chapters explain the technical machinery of the HTTP protocol, applications, and resources in much more detail.

For More Information

Later chapters of this book will explore HTTP in much more detail, but you might find that some of the following sources contain useful background about particular topics we covered in this chapter.

HTTP Protocol Information

HTTP Pocket Reference

Clinton Wong, O'Reilly & Associates, Inc. This little book provides a concise introduction to HTTP and a quick reference to each of the headers and status codes that compose HTTP transactions.

http://www.w3.org/Protocols/

This W3C web page contains many great links about the HTTP protocol.

http://www.ietf.org/rfc/rfc2616.txt

RFC 2616, "Hypertext Transfer Protocol—HTTP/1.1," is the official specification for HTTP/1.1, the current version of the HTTP protocol. The specification is a well-written, well-organized, detailed reference for HTTP, but it isn't ideal for readers who want to learn the underlying concepts and motivations of HTTP or the differences between theory and practice. We hope that this book fills in the underlying concepts, so you can make better use of the specification.

http://www.ietf.org/rfc/rfc1945.txt

RFC 1945, "Hypertext Transfer Protocol—HTTP/1.0," is an informational RFC that describes the modern foundation for HTTP. It details the officially sanctioned and "best-practice" behavior of web applications at the time the specification was written. It also contains some useful descriptions about behavior that is deprecated in HTTP/1.1 but still widely implemented by legacy applications.

http://www.w3.org/Protocols/HTTP/AsImplemented.html

This web page contains a description of the 1991 HTTP/0.9 protocol, which implements only GET requests and has no content typing.

Google Exhibit No. 1006
Page 029

Google Exhibit No. 1063
Page 0430

Historical Perspective

http://www.w3.org/Protocols/WhyHTTP.html
: This brief web page from 1991, from the author of HTTP, highlights some of the original, minimalist goals of HTTP.

http://www.w3.org/History.html
: "A Little History of the World Wide Web" gives a short but interesting perspective on some of the early goals and foundations of the World Wide Web and HTTP.

http://www.w3.org/DesignIssues/Architecture.html
: "Web Architecture from 50,000 Feet" paints a broad, ambitious view of the World Wide Web and the design principles that affect HTTP and related web technologies.

Other World Wide Web Information

http://www.w3.org
: The World Wide Web Consortium (W3C) is the technology steering team for the Web. The W3C develops interoperable technologies (specifications, guidelines, software, and tools) for the evolving Web. The W3C site is a treasure trove of introductory and detailed documentation about web technologies.

http://www.ietf.org/rfc/rfc2396.txt
: RFC 2396, "Uniform Resource Identifiers (URI): Generic Syntax," is the detailed reference for URIs and URLs.

http://www.ietf.org/rfc/rfc2141.txt
: RFC 2141, "URN Syntax," is a 1997 specification describing URN syntax.

http://www.ietf.org/rfc/rfc2046.txt
: RFC 2046, "MIME Part 2: Media Types," is the second in a suite of five Internet specifications defining the Multipurpose Internet Mail Extensions standard for multimedia content management.

http://www.wrec.org/Drafts/draft-ietf-wrec-taxonomy-06.txt
: This Internet draft, "Internet Web Replication and Caching Taxonomy," specifies standard terminology for web architectural components.

Google Exhibit No. 1006
Page 030

Google Exhibit No. 1063
Page 0431

CHAPTER 2

URLs and Resources

Think of the Internet as a giant, expanding city, full of places to see and things to do. You and the other residents and tourists of this booming community would use standard naming conventions for the city's vast attractions and services. You'd use street addresses for museums, restaurants, and people's homes. You'd use phone numbers for the fire department, the boss's secretary, and your mother, who says you don't call enough.

Everything has a standardized name, to help sort out the city's resources. Books have ISBN numbers, buses have route numbers, bank accounts have account numbers, and people have social security numbers. Tomorrow you will meet your business partners at gate 31 of the airport. Every morning you take a Red-line train and exit at Kendall Square station.

And because everyone agreed on standards for these different names, we can easily share the city's treasures with each other. You can tell the cab driver to take you to 246 McAllister Street, and he'll know what you mean (even if he takes the long way).

Uniform resource locators (URLs) are the standardized names for the Internet's resources. URLs point to pieces of electronic information, telling you where they are located and how to interact with them.

In this chapter, we'll cover:

- URL syntax and what the various URL components mean and do
- URL shortcuts that many web clients support, including relative URLs and expandomatic URLs
- URL encoding and character rules
- Common URL schemes that support a variety of Internet information systems
- The future of URLs, including uniform resource names (URNs)—a framework to support objects that move from place to place while retaining stable names

Google Exhibit No. 1006
Page 031

Google Exhibit No. 1063
Page 0432

Navigating the Internet's Resources

URLs are the resource locations that your browser needs to find information. They let people and applications find, use, and share the billions of data resources on the Internet. URLs are the usual human access point to HTTP and other protocols: a person points a browser at a URL and, behind the scenes, the browser sends the appropriate protocol messages to get the resource that the person wants.

URLs actually are a subset of a more general class of resource identifier called a uniform resource identifier, or URI. URIs are a general concept comprised of two main subsets, URLs and URNs. URLs identify resources by describing where resources are located, whereas URNs (which we'll cover later in this chapter) identify resources by name, regardless of where they currently reside.

The HTTP specification uses the more general concept of URIs as its resource identifiers; in practice, however, HTTP applications deal only with the URL subset of URIs. Throughout this book, we'll sometimes refer to URIs and URLs interchangeably, but we're almost always talking about URLs.

Say you want to fetch the URL *http://www.joes-hardware.com/seasonal/index-fall.html*:

- The first part of the URL (*http*) is the *URL scheme*. The scheme tells a web client *how* to access the resource. In this case, the URL says to use the HTTP protocol.
- The second part of the URL (*www.joes-hardware.com*) is the server location. This tells the web client *where* the resource is hosted.
- The third part of the URL (*/seasonal/index-fall.html*) is the resource path. The path tells *what* particular local resource on the server is being requested.

See Figure 2-1 for an illustration.

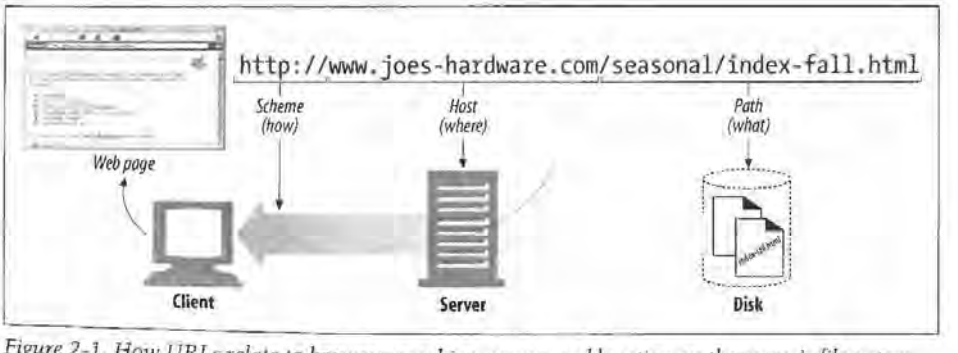


Figure 2-1. How URLs relate to browser, machine, server, and location on the server's filesystem

URLs can direct you to resources available through protocols other than HTTP. They can point you to any resource on the Internet, from a person's email account:

```
mailto:president@whitehouse.gov
```


Google Exhibit No. 1006
Page 032

Google Exhibit No. 1063
Page 0433

to files that are available through other protocols, such as the File Transfer Protocol (FTP):

```
ftp://ftp.lots-o-books.com/pub/complete-price-list.xls
```

to movies hosted off of streaming video servers:

```
rtsp://www.joes-hardware.com:554/interview/cto_video
```

URLs provide a way to uniformly name resources. Most URLs have the same "scheme://server location/path" structure. So, for every resource out there and every way to get those resources, you have a single way to name each resource so that anyone can use that name to find it. However, this wasn't always the case.

The Dark Days Before URLs

Before the Web and URLs, people relied on a rag-tag assortment of applications to access data distributed throughout the Net. Most people were not lucky enough to have all the right applications or were not savvy and patient enough to use them.

Before URLs came along, if you wanted to share the *complete-catalog.xls* file with a friend, you would have had to say something like this: "Use FTP to connect to *ftp.joes-hardware.com*. Log in as *anonymous*. Then type your username as the password. Change to the *pub* directory. Switch to binary mode. Now download the file named *complete-catalog.xls* to your local filesystem and view it there."

Today, browsers such as Netscape Navigator and Microsoft Internet Explorer bundle much of this functionality into one convenient package. Using URLs, these applications are able to access many resources in a uniform way, through one interface. Instead of the complicated instructions above, you could just say "Point your browser at *ftp://ftp.lots-o-books.com/pub/complete-catalog.xls*."

URLs have provided a means for applications to be aware of how to access a resource. In fact, many users are probably unaware of the protocols and access methods their browsers use to get the resources they are requesting.

With web browsers, you no longer need a news reader to read Internet news or an FTP client to access files on FTP servers. You don't need an electronic mail program to send and receive email messages. URLs have helped to simplify the online world, by allowing the browser to be smart about how to access and handle resources.* Applications can use URLs to simplify access to information.

URLs give you and your browser all you need to find a piece of information. They define the particular resource you want, where it is located, and how to get it.

* Browsers often use other applications to handle specific resources. For example, Internet Explorer launches an email application to handle URLs that identify email resources.

Google Exhibit No. 1006
Page 033

Google Exhibit No. 1063
Page 0434

URL Syntax

URLs provide a means of locating any resource on the Internet, but these resources can be accessed by different schemes (e.g., HTTP, FTP, SMTP), and URL syntax varies from scheme to scheme.

Does this mean that each different URL scheme has a radically different syntax? In practice, no. Most URLs adhere to a general URL syntax, and there is significant overlap in the style and syntax between different URL schemes.

Most URL schemes base their URL syntax on this nine-part general format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

Almost no URLs contain all these components. The three most important parts of a URL are the *scheme*, the *host*, and the *path*. Table 2-1 summarizes the various components.

Table 2-1. General URL components

Component	Description	Default value
scheme	Which protocol to use when accessing a server to get a resource.	None
user	The username some schemes require to access a resource.	anonymous
password	The password that may be included after the username, separated by a colon (:).	<Email address>
host	The hostname or dotted IP address of the server hosting the resource.	None
port	The port number on which the server hosting the resource is listening. Many schemes have default port numbers (the default port number for HTTP is 80).	Scheme-specific
path	The local name for the resource on the server, separated from the previous URL components by a slash (/). The syntax of the path component is server- and scheme-specific. (We will see later in this chapter that a URL's path can be divided into segments, and each segment can have its own components specific to that segment.)	None
params	Used by some schemes to specify input parameters. Params are name/value pairs. A URL can contain multiple params fields, separated from themselves and the rest of the path by semicolons (;).	None
query	Used by some schemes to pass parameters to active applications (such as databases, bulletin boards, search engines, and other Internet gateways). There is no common format for the contents of the query component. It is separated from the rest of the URL by the "?" character.	None
frag	A name for a piece or part of the resource. The frag field is not passed to the server when referencing the object; it is used internally by the client. It is separated from the rest of the URL by the "#" character.	None

For example, consider the URL *http://www.joes-hardware.com:80/index.html*. The scheme is "http", the host is "www.joes-hardware.com", the port is "80", and the path is "/index.html".

Google Exhibit No. 1006
Page 034

Google Exhibit No. 1063
Page 0435

Schemes: What Protocol to Use

The *scheme* is really the main identifier of how to access a given resource; it tells the application interpreting the URL what protocol it needs to speak. In our simple HTTP URL, the scheme is simply "http".

The scheme component must start with an alphabetic character, and it is separated from the rest of the URL by the first ":" character. Scheme names are case-insensitive, so the URLs "http://www.joes-hardware.com" and "HTTP://www.joes-hardware.com" are equivalent.

Hosts and Ports

To find a resource on the Internet, an application needs to know what machine is hosting the resource and where on that machine it can find the server that has access to the desired resource. The *host* and *port* components of the URL provide these two pieces of information.

The host component identifies the host machine on the Internet that has access to the resource. The name can be provided as a hostname, as above ("www.joes-hardware.com") or as an IP address. For example, the following two URLs point to the same resource—the first refers to the server by its hostname and the second by its IP address:

```
http://www.joes-hardware.com:80/index.html
http://161.58.228.45:80/index.html
```

The port component identifies the network port on which the server is listening. For HTTP, which uses the underlying TCP protocol, the default port is 80.

Usernames and Passwords

More interesting components are the *user* and *password* components. Many servers require a username and password before you can access data through them. FTP servers are a common example of this. Here are a few examples:

```
ftp://ftp.prep.ai.mit.edu/pub/gnu
ftp://anonymous@ftp.prep.ai.mit.edu/pub/gnu
ftp://anonymous:my_passwd@ftp.prep.ai.mit.edu/pub/gnu
http://joe:joespasswd@www.joes-hardware.com/sales_info.txt
```

The first example has no user or password component, just our standard scheme, host, and path. If an application is using a URL scheme that requires a username and password, such as FTP, it generally will insert a default username and password if they aren't supplied. For example, if you hand your browser an FTP URL without specifying a username and password, it will insert "anonymous" for your username and send a default password (Internet Explorer sends "IEUser", while Netscape Navigator sends "mozilla").

Google Exhibit No. 1006
Page 035

Google Exhibit No. 1063
Page 0436

The second example shows a username being specified as "anonymous". This username, combined with the host component, looks just like an email address. The "@" character separates the user and password components from the rest of the URL.

In the third example, both a username ("anonymous") and password ("my_passwd") are specified, separated by the ":" character.

Paths

The *path* component of the URL specifies where on the server machine the resource lives. The path often resembles a hierarchical filesystem path. For example:

```
http://www.joes-hardware.com:80/seasonal/index-fall.html
```

The path in this URL is "/seasonal/index-fall.html", which resembles a filesystem path on a Unix filesystem. The path is the information that the server needs to locate the resource.* The path component for HTTP URLs can be divided into *path segments* separated by "/" characters (again, as in a file path on a Unix filesystem). Each path segment can have its own *params* component.

Parameters

For many schemes, a simple host and path to the object just aren't enough. Aside from what port the server is listening to and even whether or not you have access to the resource with a username and password, many protocols require more information to work.

Applications interpreting URLs need these protocol parameters to access the resource. Otherwise, the server on the other side might not service the request or, worse yet, might service it wrong. For example, take a protocol like FTP, which has two modes of transfer, binary and text. You wouldn't want your binary image transferred in text mode, because the binary image could be scrambled.

To give applications the input parameters they need in order to talk to the server correctly, URLs have a *params* component. This component is just a list of name/value pairs in the URL, separated from the rest of the URL (and from each other) by ";" characters. They provide applications with any additional information that they need to access the resource. For example:

```
ftp://prep.ai.mit.edu/pub/gnu;type=d
```

In this example, there is one param, `type=d`, where the name of the param is "type" and its value is "d".

* This is a bit of a simplification. In "Virtual Hosting" in Chapter 18, we will see that the path is not always enough information to locate a resource. Sometimes a server needs additional information.

Google Exhibit No. 1006
Page 036

Google Exhibit No. 1063
Page 0437

As we mentioned earlier, the path component for HTTP URLs can be broken into path segments. Each segment can have its own params. For example:

```
http://www.joes-hardware.com/hammers;sale=false/index.html;graphics=true
```

In this example there are two path segments, `hammers` and `index.html`. The `hammers` path segment has the param `sale`, and its value is `false`. The `index.html` segment has the param `graphics`, and its value is `true`.

Query Strings

Some resources, such as database services, can be asked questions or queries to narrow down the type of resource being requested.

Let's say Joe's Hardware store maintains a list of unsold inventory in a database and allows the inventory to be queried, to see whether products are in stock. The following URL might be used to query a web database gateway to see if item number 12731 is available:

```
http://www.joes-hardware.com/inventory-check.cgi?item=12731
```

For the most part, this resembles the other URLs we have looked at. What is new is everything to the right of the question mark (?). This is called the *query* component. The query component of the URL is passed along to a gateway resource, with the path component of the URL identifying the gateway resource. Basically, gateways can be thought of as access points to other applications (we discuss gateways in detail in Chapter 8).

Figure 2-2 shows an example of a query component being passed to a server that is acting as a gateway to Joe's Hardware's inventory-checking application. The query is checking whether a particular item, `12731`, is in inventory in size large and color `blue`.

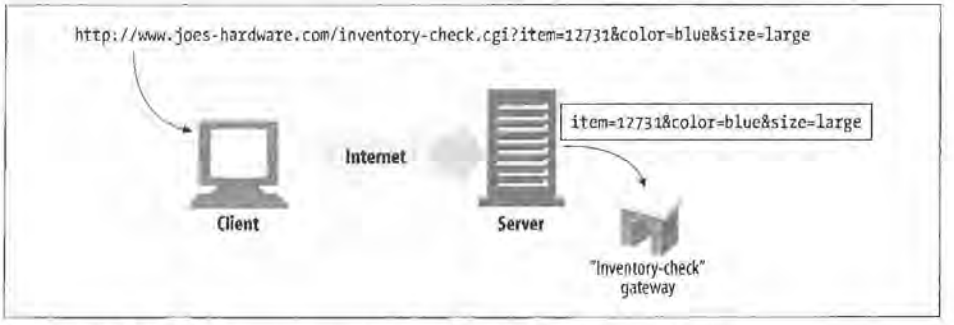


Figure 2-2. The URL query component is sent along to the gateway application

There is no requirement for the format of the query component, except that some characters are illegal, as we'll see later in this chapter. By convention, many gateways

Google Exhibit No. 1006
Page 037

Google Exhibit No. 1063
Page 0438

expect the query string to be formatted as a series of "name=value" pairs, separated by "&" characters:

```
http://www.joes-hardware.com/inventory-check.cgi?item=12731&color=blue
```

In this example, there are two name/value pairs in the query component: `item=12731` and `color=blue`.

Fragments

Some resource types, such as HTML, can be divided further than just the resource level. For example, for a single, large text document with sections in it, the URL for the resource would point to the entire text document, but ideally you could specify the sections within the resource.

To allow referencing of parts or fragments of a resource, URLs support a *frag* component to identify pieces within a resource. For example, a URL could point to a particular image or section within an HTML document.

A fragment dangles off the right-hand side of a URL, preceded by a # character. For example:

```
http://www.joes-hardware.com/tools.html#drills
```

In this example, the fragment `drills` references a portion of the */tools.html* web page located on the Joe's Hardware web server. The portion is named "drills".

Because HTTP servers generally deal only with entire objects,* not with fragments of objects, clients don't pass fragments along to servers (see Figure 2-3). After your browser gets the *entire* resource from the server, it then uses the fragment to display the part of the resource in which you are interested.

URL Shortcuts

Web clients understand and use a few URL shortcuts. Relative URLs are a convenient shorthand for specifying a resource within a resource. Many browsers also support "automatic expansion" of URLs, where the user can type in a key (memorable) part of a URL, and the browser fills in the rest. This is explained in "Expandomatic URLs."

Relative URLs

URLs come in two flavors: *absolute* and *relative*. So far, we have looked only at absolute URLs. With an absolute URL, you have all the information you need to access a resource.

* In "Range Requests" in Chapter 15, we will see that HTTP agents may request byte ranges of objects. However, in the context of URL fragments, the server sends the entire object and the agent applies the fragment identifier to the resource.

Google Exhibit No. 1006
Page 038

Google Exhibit No. 1063
Page 0439

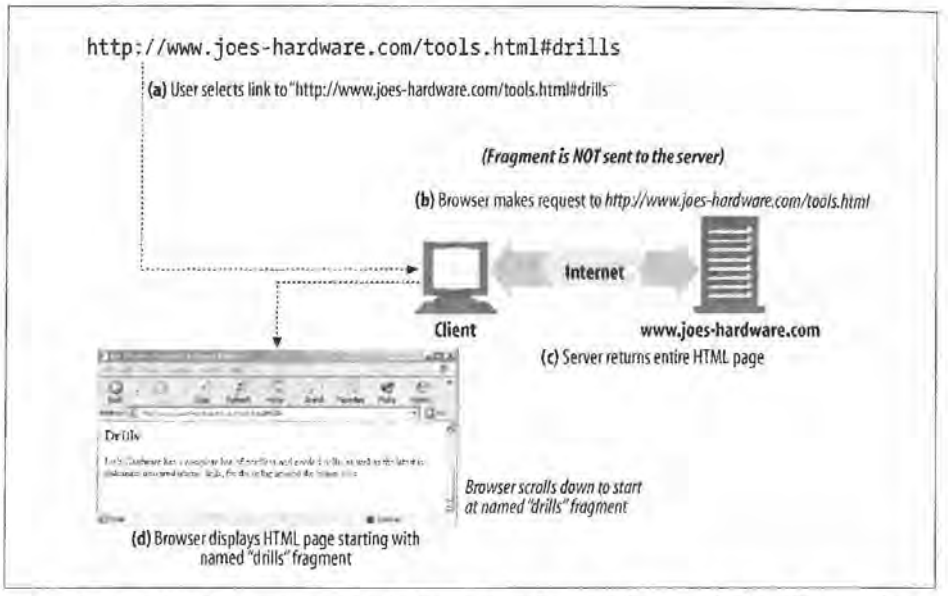


Figure 2-3. The URL fragment is used only by the client, because the server deals with entire objects

On the other hand, relative URLs are incomplete. To get all the information needed to access a resource from a relative URL, you must interpret it relative to another URL, called its *base*.

Relative URLs are a convenient shorthand notation for URLs. If you have ever written HTML by hand, you have probably found them to be a great shortcut. Example 2-1 contains an example HTML document with an embedded relative URL.

Example 2-1. HTML snippet with relative URLs

```
<HTML>
<HEAD><TITLE>Joe's Tools</TITLE></HEAD>
<BODY>
<H1> Tools Page </H1>
<H2> Hammers <H2>
<P> Joe's Hardware Online has the largest selection of <A HREF="./hammers.html">hammers
</A> on earth.
</BODY>
</HTML>
```

In Example 2-1, we have an HTML document for the resource:

```
http://www.joes-hardware.com/tools.html
```

In the HTML document, there is a hyperlink containing the URL *./hammers.html*. This URL seems incomplete, but it is a legal relative URL. It can be interpreted relative to the URL of the document in which it is found; in this case, relative to the resource */tools.html* on the Joe's Hardware web server.

Google Exhibit No. 1006
Page 039

Google Exhibit No. 1063
Page 0440

The abbreviated relative URL syntax lets HTML authors omit from URLs the scheme, host, and other components. These components can be inferred by the *base* URL of the resource they are in. URLs for other resources also can be specified in this shorthand.

In Example 2-1, our base URL is:

```
http://www.joes-hardware.com/tools.html
```

Using this URL as a base, we can infer the missing information. We know the resource is *./hammers.html*, but we don't know the scheme or host. Using the base URL, we can infer that the scheme is *http* and the host is *www.joes-hardware.com*. Figure 2-4 illustrates this.



Figure 2-4. Using a base URL

Relative URLs are only fragments or pieces of URLs. Applications that process URLs (such as your browser) need to be able to convert between relative and absolute URLs.

It is also worth noting that relative URLs provide a convenient way to keep a set of resources (such as HTML pages) portable. If you use relative URLs, you can move a set of documents around and still have their links work, because they will be interpreted relative to the new base. This allows for things like mirroring content on other servers.

Base URLs

The first step in the conversion process is to find a base URL. The base URL serves as a point of reference for the relative URL. It can come from a few places:

Explicitly provided in the resource
: Some resources explicitly specify the base URL. An HTML document, for example, may include a `<BASE>` HTML tag defining the base URL by which to convert all relative URLs in that HTML document.

Base URL of the encapsulating resource
: If a relative URL is found in a resource that does not explicitly specify a base URL, as in Example 2-1, it can use the URL of the resource in which it is embedded as a base (as we did in our example).

Google Exhibit No. 1006
Page 040

Google Exhibit No. 1063
Page 0441

No base URL

In some instances, there is no base URL. This often means that you have an absolute URL; however, sometimes you may just have an incomplete or broken URL.

Resolving relative references

Previously, we showed the basic components and syntax of URLs. The next step in converting a relative URL into an absolute one is to break up both the relative and base URLs into their component pieces.

In effect, you are just parsing the URL, but this is often called *decomposing* the URL, because you are breaking it up into its components. Once you have broken the base and relative URLs into their components, you can then apply the algorithm pictured in Figure 2-5 to finish the conversion.

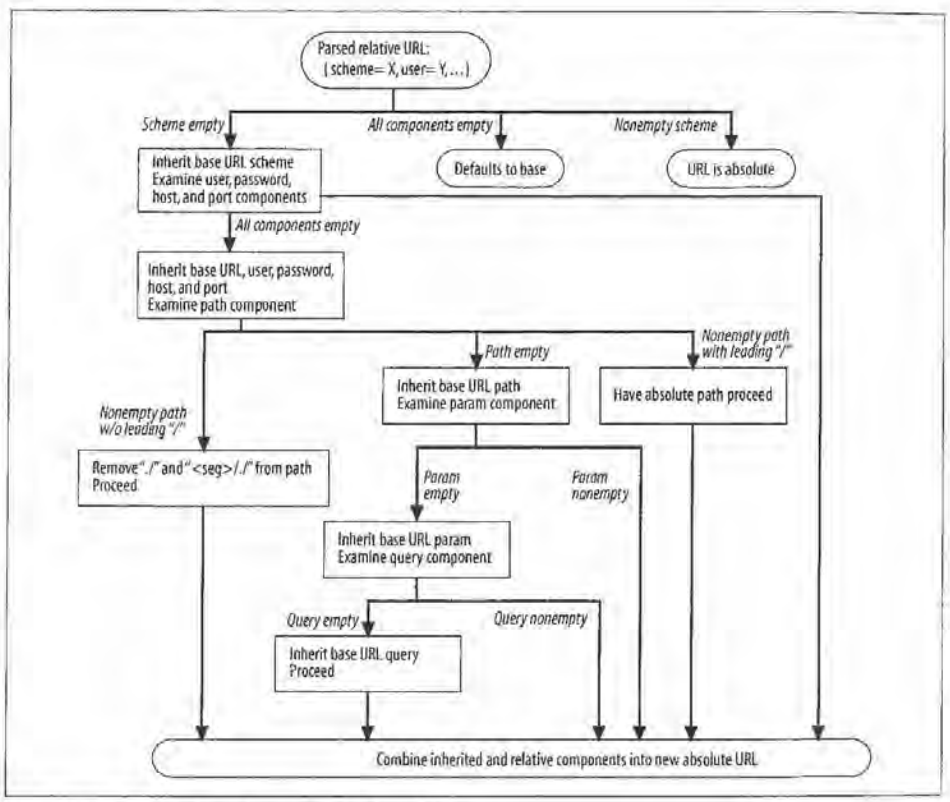


Figure 2-5. Converting relative to absolute URLs

Google Exhibit No. 1006
Page 041

Google Exhibit No. 1063
Page 0442

This algorithm converts a relative URL to its absolute form, which can then be used to reference the resource. This algorithm was originally specified in RFC 1808 and later incorporated into RFC 2396.

With our *./hammers.html* example from Example 2-1, we can apply the algorithm depicted in Figure 2-5:

1. Path is *./hammers.html*; base URL is *http://www.joes-hardware.com/tools.html*.
2. Scheme is empty; proceed down left half of chart and inherit the base URL scheme (HTTP).
3. At least one component is non-empty; proceed to bottom, inheriting host and port components.
4. Combining the components we have from the relative URL (path: *./hammers.html*) with what we have inherited (scheme: *http*, host: *www.joes-hardware.com*, port: *80*), we get our new absolute URL: *http://www.joes-hardware.com/hammers.html*.

Expandomatic URLs

Some browsers try to expand URLs automatically, either after you submit the URL or while you're typing. This provides users with a shortcut: they don't have to type in the complete URL, because it automatically expands itself.

These "expandomatic" features come in two flavors:

Hostname expansion

In hostname expansion, the browser can often expand the hostname you type in into the full hostname without your help, just by using some simple heuristics.

For example if you type "yahoo" in the address box, your browser can automatically insert "www." and ".com" onto the hostname, creating "www.yahoo.com". Some browsers will try this if they are unable to find a site that matches "yahoo", trying a few expansions before giving up. Browsers apply these simple tricks to save you some time and frustration.

However, these expansion tricks on hostnames can cause problems for other HTTP applications, such as proxies. In Chapter 6, we will discuss these problems in more detail.

History expansion

Another technique that browsers use to save you time typing URLs is to store a history of the URLs that you have visited in the past. As you type in the URL, they can offer you completed choices to select from by matching what you type to the prefixes of the URLs in your history. So, if you were typing in the start of a URL that you had visited previously, such as *http://www.joes-*, your browser could suggest *http://www.joes-hardware.com*. You could then select that instead of typing out the complete URL.

Google Exhibit No. 1006
Page 042

Google Exhibit No. 1063
Page 0443

Be aware that URL auto-expansion may behave differently when used with proxies. We discuss this further in "URI Client Auto-Expansion and Hostname Resolution" in Chapter 6.

Shady Characters

URLs were designed to be *portable*. They were also designed to uniformly name all the resources on the Internet, which means that they will be transmitted through various protocols. Because all of these protocols have different mechanisms for transmitting their data, it was important for URLs to be designed so that they could be transmitted safely through any Internet protocol.

Safe transmission means that URLs can be transmitted without the risk of losing information. Some protocols, such as the Simple Mail Transfer Protocol (SMTP) for electronic mail, use transmission methods that can strip off certain characters.* To get around this, URLs are permitted to contain only characters from a relatively small, universally safe alphabet.

In addition to wanting URLs to be transportable by all Internet protocols, designers wanted them to be *readable* by people. So invisible, nonprinting characters also are prohibited in URLs, even though these characters may pass through mailers and otherwise be portable.†

To complicate matters further, URLs also need to be *complete*. URL designers realized there would be times when people would want URLs to contain binary data or characters outside of the universally safe alphabet. So, an escape mechanism was added, allowing unsafe characters to be encoded into safe characters for transport.

This section summarizes the universal alphabet and encoding rules for URLs.

The URL Character Set

Default computer system character sets often have an Anglocentric bias. Historically, many computer applications have used the US-ASCII character set. US-ASCII uses 7 bits to represent most keys available on an English typewriter and a few nonprinting control characters for text formatting and hardware signalling.

US-ASCII is very portable, due to its long legacy. But while it's convenient to citizens of the U.S., it doesn't support the inflected characters common in European languages or the hundreds of non-Romanic languages read by billions of people around the world.

* This is caused by the use of a 7-bit encoding for messages; this can strip off information if the source is encoded in 8 bits or more.

† Nonprinting characters include whitespace (note that RFC 2396 recommends that applications ignore whitespace).

Google Exhibit No. 1006
Page 043

Google Exhibit No. 1063
Page 0444

Furthermore, some URLs may need to contain arbitrary binary data. Recognizing the need for completeness, the URL designers have incorporated *escape sequences*. Escape sequences allow the encoding of arbitrary character values or data using a restricted subset of the US-ASCII character set, yielding portability and completeness.

Encoding Mechanisms

To get around the limitations of a safe character set representation, an encoding scheme was devised to represent characters in a URL that are not safe. The encoding simply represents the unsafe character by an "escape" notation, consisting of a percent sign (%) followed by two hexadecimal digits that represent the ASCII code of the character.

Table 2-2 shows a few examples.

Table 2-2. Some encoded character examples

Character	ASCII code	Example URL
~	126 (0x7E)	*http://www.joes-hardware.com/%7Ejoe*
SPACE	32 (0x20)	*http://www.joes-hardware.com/more%20tools.html*
%	37 (0x25)	*http://www.joes-hardware.com/100%25satisfaction.html*

Character Restrictions

Several characters have been reserved to have special meaning inside of a URL. Others are not in the defined US-ASCII printable set. And still others are known to confuse some Internet gateways and protocols, so their use is discouraged.

Table 2-3 lists characters that should be encoded in a URL before you use them for anything other than their reserved purposes.

Table 2-3. Reserved and restricted characters

Character	Reservation/Restriction
%	Reserved as escape token for encoded characters
/	Reserved for delimiting splitting up path segments in the path component
.	Reserved in the path component
..	Reserved in the path component
#	Reserved as the fragment delimiter
?	Reserved as the query-string delimiter
;	Reserved as the params delimiter
:	Reserved to delimit the scheme, user/password, and host/port components
$, +	Reserved
@ & =	Reserved because they have special meaning in the context of some schemes

Google Exhibit No. 1006
Page 044

Google Exhibit No. 1063
Page 0445

Table 2-3. Reserved and restricted characters (continued)

Character	Reservation/Restriction
{}|\^~[]'	Restricted because of unsafe handling by various transport agents, such as gateways
<>"	Unsafe; should be encoded because these characters often have meaning outside the scope of the URL, such as delimiting the URL itself in a document (e.g., "http://www.joes-hardware.com")
0x00–0x1F, 0x7F	Restricted; characters within these hex ranges fall within the nonprintable section of the US-ASCII character set
> 0x7F	Restricted; characters whose hex values fall within this range do not fall within the 7-bit range of the US-ASCII character set

A Bit More

You might be wondering why nothing bad has happened when you have used characters that are unsafe. For instance, you can visit Joe's home page at:

```
http://www.joes-hardware.com/~joe
```

and not encode the "~" character. For some transport protocols this is not an issue, but it is still unwise for application developers not to encode unsafe characters.

Applications need to walk a fine line. It is best for client applications to convert any unsafe or restricted characters before sending any URL to any other application.* Once all the unsafe characters have been encoded, the URL is in a *canonical form* that can be shared between applications; there is no need to worry about the other application getting confused by any of the characters' special meanings.

The original application that gets the URL from the user is best fit to determine which characters need to be encoded. Because each component of the URL may have its own safe/unsafe characters, and which characters are safe/unsafe is scheme-dependent, only the application receiving the URL from the user really is in a position to determine what needs to be encoded.

Of course, the other extreme is for the application to encode all characters. While this is not recommended, there is no hard and fast rule against encoding characters that are considered safe already; however, in practice this can lead to odd and broken behavior, because some applications may assume that safe characters will not be encoded.

Sometimes, malicious folks encode extra characters in an attempt to get around applications that are doing pattern matching on URLs—for example, web filtering applications. Encoding safe URL components can cause pattern-matching applications to fail to recognize the patterns for which they are searching. In general, applications interpreting URLs must decode the URLs before processing them.

* Here we are specifically talking about client applications, not other HTTP intermediaries, like proxies. In "In-Flight URI Modification" in Chapter 6, we discuss some of the problems that can arise when proxies or other intermediary HTTP applications attempt to change (e.g., encode) URLs on the behalf of a client.

Google Exhibit No. 1006
Page 045

Google Exhibit No. 1063
Page 0446

Some URL components, such as the scheme, need to be recognized readily and are required to start with an alphabetic character. Refer back to "URL Syntax" for more guidelines on the use of reserved and unsafe characters within different URL components.*

A Sea of Schemes

In this section, we'll take a look at the more common scheme formats on the Web. Appendix A gives a fairly exhaustive list of schemes and references to their individual documentation.

Table 2-4 summarizes some of the most popular schemes. Reviewing "URL Syntax" will make the syntax portion of the table a little more familiar.

Table 2-4. Common scheme formats

Scheme	Description
http	The Hypertext Transfer Protocol scheme conforms to the general URL format, except that there is no username or password. The port defaults to 80 if omitted. Basic form: *http://<host>:<port>/<path>?<query>#<frag>* Examples: *http://www.joes-hardware.com/index.html* *http://www.joes-hardware.com:80/index.html*
https	The https scheme is a twin to the http scheme. The only difference is that the https scheme uses Netscape's Secure Sockets Layer (SSL), which provides end-to-end encryption of HTTP connections. Its syntax is identical to that of HTTP, with a default port of 443. Basic form: *https://<host>:<port>/<path>?<query>#<frag>* Example: *https://www.joes-hardware.com/secure.html*
mailto	Mailto URLs refer to email addresses. Because email behaves differently from other schemes (it does not refer to objects that can be accessed directly), the format of a mailto URL differs from that of the standard URL. The syntax for Internet email addresses is documented in Internet RFC 822. Basic form: *mailto:<RFC-822-addr-spec>* Example: *mailto:joe@joes-hardware.com*

* Table 2-3 lists reserved characters for the various URL components. In general, encoding should be limited to those characters that are unsafe for transport.

Google Exhibit No. 1006
Page 046

Google Exhibit No. 1063
Page 0447

Table 2-4. Common scheme formats (continued)

Scheme	Description
ftp	File Transfer Protocol URLs can be used to download and upload files on an FTP server and to obtain listings of the contents of a directory structure on an FTP server. FTP has been around since before the advent of the Web and URLs. Web applications have assimilated FTP as a data-access scheme. The URL syntax follows the general form. Basic form: *ftp://<user>:<password>@<host>:<port>/<path>;<params>* Example: *ftp://anonymous:joe%40joes-hardware.com@prep.ai.mit.edu:21/pub/gnu/*
rtsp, rtspu	RTSP URLs are identifiers for audio and video media resources that can be retrieved through the Real Time Streaming Protocol. The "u" in the rtspu scheme denotes that the UDP protocol is used to retrieve the resource. Basic forms: *rtsp://<user>:<password>@<host>:<port>/<path>* *rtspu://<user>:<password>@<host>:<port>/<path>* Example: *rtsp://www.joes-hardware.com:554/interview/cto_video*
file	The file scheme denotes files directly accessible on a given host machine (by local disk, a network filesystem, or some other file-sharing system). The fields follow the general form. If the host is omitted, it defaults to the local host from which the URL is being used. Basic form: *file://<host>/<path>* Example: *file://OFFICE-FS/policies/casual-fridays.doc*
news	The news scheme is used to access specific articles or newsgroups, as defined by RFC 1036. It has the unusual property that a news URL in itself does not contain sufficient information to locate the resource. The news URL is missing information about where to acquire the resource—no hostname or machine name is supplied. It is the interpreting application's job to acquire this information from the user. For example, in your Netscape browser, under the Options menu, you can specify your NNTP (news) server. This tells your browser what server to use when it has a news URL. News resources can be accessed from multiple servers. They are said to be location-independent, as they are not dependent on any one source for access. The "@" character is reserved within a news URL and is used to distinguish between news URLs that refer to newsgroups and news URLs that refer to specific news articles. Basic forms: *news:<newsgroup>* *news:<news-article-id>* Example: *news:rec.arts.startrek*

Google Exhibit No. 1006
Page 047

Google Exhibit No. 1063
Page 0448

Table 2-4. Common scheme formats (continued)

Scheme	Description
telnet	The telnet scheme is used to access interactive services. It does not represent an object per se, but an interactive application (resource) accessible via the telnet protocol. Basic form: *telnet://<user>:<password>@<host>:<port>/* Example: *telnet://slurp:webhound@joes-hardware.com:23/*

The Future

URLs are a powerful tool. Their design allows them to name all existing objects and easily encompass new formats. They provide a uniform naming mechanism that can be shared between Internet protocols.

However, they are not perfect. URLs are really addresses, not true names. This means that a URL tells you where something is located, for the moment. It provides you with the name of a specific server on a specific port, where you can find the resource. The downfall of this scheme is that if the resource is moved, the URL is no longer valid. And at that point, it provides no way to locate the object.

What would be ideal is if you had the real name of an object, which you could use to look up that object regardless of its location. As with a person, given the name of the resource and a few other facts, you could track down that resource, regardless of where it moved.

The Internet Engineering Task Force (IETF) has been working on a new standard, uniform resource names (URNs), for some time now, to address just this issue. URNs provide a stable name for an object, regardless of where that object moves (either inside a web server or across web servers).

Persistent uniform resource locators (PURLs) are an example of how URN functionality can be achieved using URLs. The concept is to introduce another level of indirection in looking up a resource, using an intermediary *resource locator* server that catalogues and tracks the actual URL of a resource. A client can request a persistent URL from the locator, which can then respond with a resource that redirects the client to the actual and current URL for the resource (see Figure 2-6). For more information on PURLs, visit *http://purl.oclc.org*.

If Not Now, When?

The ideas behind URNs have been around for some time. Indeed, if you look at the publication dates for some of their specifications, you might ask yourself why they have yet to be adopted.

Google Exhibit No. 1006
Page 048

Google Exhibit No. 1063
Page 0449

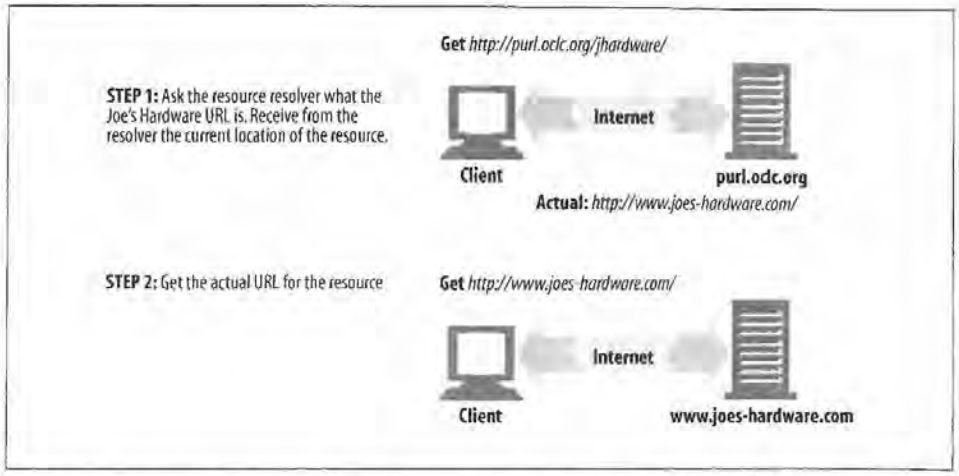


Figure 2-6. PURLs use a resource locator server to name the current location of a resource

The change from URLs to URNs is an enormous task. Standardization is a slow process, often for good reason. Support for URNs will require many changes—consensus from the standards bodies, modifications to various HTTP applications, etc. A tremendous amount of critical mass is required to make such changes, and unfortunately (or perhaps fortunately), there is so much momentum behind URLs that it will be some time before all the stars align to make such a conversion possible.

Throughout the explosive growth of the Web, Internet users—everyone from computer scientists to the average Internet user—have been taught to use URLs. While they suffer from clumsy syntax (for the novice) and persistence problems, people have learned how to use them and how to deal with their drawbacks. URLs have some limitations, but they're not the web development community's most pressing problem.

Currently, and for the foreseeable future, URLs are the way to name resources on the Internet. They are everywhere, and they have proven to be a very important part of the Web's success. It will be a while before any other naming scheme unseats URLs. However, URLs do have their limitations, and it is likely that new standards (possibly URNs) will emerge and be deployed to address some of these limitations.

For More Information

For more information on URLs, refer to:

http://www.w3.org/Addressing/
: The W3C web page about naming and addressing URIs and URLs.

http://www.ietf.org/rfc/rfc1738
: RFC 1738, "Uniform Resource Locators (URL)," by T. Berners-Lee, L. Masinter, and M. McCahill.

Google Exhibit No. 1006
Page 049

Google Exhibit No. 1063
Page 0450

http://www.ietf.org/rfc/rfc2396.txt
RFC 2396, "Uniform Resource Identifiers (URI): Generic Syntax," by T. Berners-Lee, R. Fielding, and L. Masinter.

http://www.ietf.org/rfc/rfc2141.txt
RFC 2141, "URN Syntax," by R. Moats.

http://purl.oclc.org
The persistent uniform resource locator web site.

http://www.ietf.org/rfc/rfc1808.txt
RFC 1808, "Relative Uniform Resource Locators," by R. Fielding.

Google Exhibit No. 1006
Page 050

Google Exhibit No. 1063
Page 0451

CHAPTER 5

Web Servers

Web servers dish out billions of web pages a day. They tell you the weather, load up your online shopping carts, and let you find long-lost high-school buddies. Web servers are the workhorses of the World Wide Web. In this chapter, we:

- Survey the many different types of software and hardware web servers.
- Describe how to write a simple diagnostic web server in Perl.
- Explain how web servers process HTTP transactions, step by step.

Where it helps to make things concrete, our examples use the Apache web server and its configuration options.

Web Servers Come in All Shapes and Sizes

A web server processes HTTP requests and serves responses. The term "web server" can refer either to web server software or to the particular device or computer dedicated to serving the web pages.

Web servers comes in all flavors, shapes, and sizes. There are trivial 10-line Perl script web servers, 50-MB secure commerce engines, and tiny servers-on-a-card. But whatever the functional differences, all web servers receive HTTP requests for resources and serve content back to the clients (look back to Figure 1-5).

Web Server Implementations

Web servers implement HTTP and the related TCP connection handling. They also manage the resources served by the web server and provide administrative features to configure, control, and enhance the web server.

The web server logic implements the HTTP protocol, manages web resources, and provides web server administrative capabilities. The web server logic shares responsibilities for managing TCP connections with the operating system. The underlying

Google Exhibit No. 1006
Page 051

Google Exhibit No. 1063
Page 0452

operating system manages the hardware details of the underlying computer system and provides TCP/IP network support, filesystems to hold web resources, and process management to control current computing activities.

Web servers are available in many forms:

- You can install and run general-purpose software web servers on standard computer systems.
- If you don't want the hassle of installing software, you can purchase a web server appliance, in which the software comes preinstalled and preconfigured on a computer, often in a snazzy-looking chassis.
- Given the miracles of microprocessors, some companies even offer embedded web servers implemented in a small number of computer chips, making them perfect administration consoles for consumer devices.

Let's look at each of those types of implementations.

General-Purpose Software Web Servers

General-purpose software web servers run on standard, network-enabled computer systems. You can choose open source software (such as Apache or W3C's Jigsaw) or commercial software (such as Microsoft's and iPlanet's web servers). Web server software is available for just about every computer and operating system.

While there are tens of thousands of different kinds of web server programs (including custom-crafted, special-purpose web servers), most web server software comes from a small number of organizations.

In February 2002, the Netcraft survey (*http://www.netcraft.com/survey/*) showed three vendors dominating the public Internet web server market (see Figure 5-1):

- The free Apache software powers nearly 60% of all Internet web servers.
- Microsoft web server makes up another 30%.
- Sun iPlanet servers comprise another 3%.

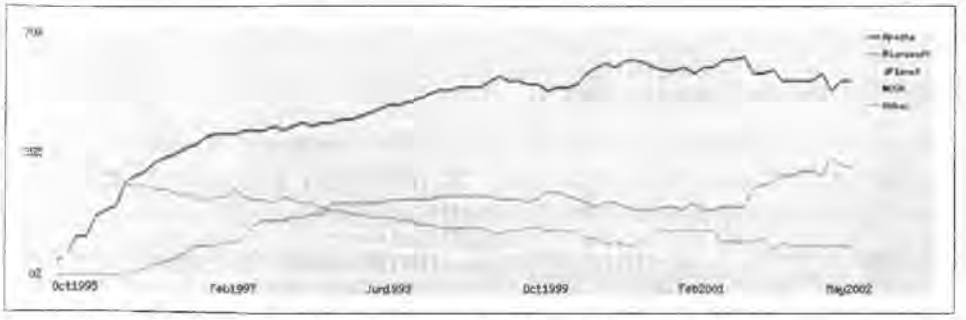


Figure 5-1. Web server market share as estimated by Netcraft's automated survey

Google Exhibit No. 1006
Page 052

Google Exhibit No. 1063
Page 0453

Take these numbers with a few grains of salt, however, as the Netcraft survey is commonly believed to exaggerate the dominance of Apache software. First, the survey counts servers independent of server popularity. Proxy server access studies from large ISPs suggest that the amount of pages served from Apache servers is much less than 60% but still exceeds Microsoft and Sun iPlanet. Additionally, it is anecdotally believed that Microsoft and iPlanet servers are more popular than Apache inside corporate enterprises.

Web Server Appliances

Web server appliances are prepackaged software/hardware solutions. The vendor preinstalls a software server onto a vendor-chosen computer platform and preconfigures the software. Some examples of web server appliances include:

- Sun/Cobalt RaQ web appliances (*http://www.cobalt.com*)
- Toshiba Magnia SG10 (*http://www.toshiba.com*)
- IBM Whistle web server appliance (*http://www.whistle.com*)

Appliance solutions remove the need to install and configure software and often greatly simplify administration. However, the web server often is less flexible and feature-rich, and the server hardware is not easily repurposeable or upgradable.

Embedded Web Servers

Embedded servers are tiny web servers intended to be embedded into consumer products (e.g., printers or home appliances). Embedded web servers allow users to administer their consumer devices using a convenient web browser interface.

Some embedded web servers can even be implemented in less than one square inch, but they usually offer a minimal feature set. Two examples of very small embedded web servers are:

- IPic match-head sized web server (*http://www-ccs.cs.umass.edu/~shri/iPic.html*)
- NetMedia SitePlayer SP1 Ethernet Web Server (*http://www.siteplayer.com*)

A Minimal Perl Web Server

If you want to build a full-featured HTTP server, you have some work to do. The core of the Apache web server has over 50,000 lines of code, and optional processing modules make that number much bigger.

All this software is needed to support HTTP/1.1 features: rich resource support, virtual hosting, access control, logging, configuration, monitoring, and performance features. That said, you can create a minimally functional HTTP server in under 30 lines of Perl. Let's take a look.

Google Exhibit No. 1006
Page 053

Google Exhibit No. 1063
Page 0454

Example 5-1 shows a tiny Perl program called *type-o-serve*. This program is a useful diagnostic tool for testing interactions with clients and proxies. Like any web server, *type-o-serve* waits for an HTTP connection. As soon as *type-o-serve* gets the request message, it prints the message on the screen; then it waits for you to type (or paste) in a response message, which is sent back to the client. This way, *type-o-serve* pretends to be a web server, records the exact HTTP request messages, and allows you to send back any HTTP response message.

This simple *type-o-serve* utility doesn't implement most HTTP functionality, but it is a useful tool to generate server response messages the same way you can use Telnet to generate client request messages (refer back to Example 5-1). You can download the *type-o-serve* program from *http://www.http-guide.com/tools/type-o-serve.pl*.

Example 5-1. type-o-serve—a minimal Perl web server used for HTTP debugging

```
#!/usr/bin/perl

use Socket;
use Carp;
use FileHandle;

# (1) use port 8080 by default, unless overridden on command line
$port = (@ARGV ? $ARGV[0] : 8080);

# (2) create local TCP socket and set it to listen for connections
$proto = getprotobyname('tcp');
socket(S, PF_INET, SOCK_STREAM, $proto) || die;
setsockopt(S, SOL_SOCKET, SO_REUSEADDR, pack("l", 1)) || die;
bind(S, sockaddr_in($port, INADDR_ANY)) || die;
listen(S, SOMAXCONN) || die;

# (3) print a startup message
printf("   <<<Type-O-Serve Accepting on Port %d>>>\n\n",$port);

while (1)
{
    # (4) wait for a connection C
    $cport_caddr = accept(C, S);
    ($cport,$caddr) = sockaddr_in($cport_caddr);
    C->autoflush(1);

    # (5) print who the connection is from
    $cname = gethostbyaddr($caddr,AF_INET);
    printf("   <<<Request From '%s'>>>\n",$cname);

    # (6) read request msg until blank line, and print on screen
    while ($line = <C>)
    {
        print $line;
        if ($line =~ /^\r/) { last; }
    }
```


Google Exhibit No. 1006
Page 054

Google Exhibit No. 1063
Page 0455

Example 5-1. type-o-serve—a minimal Perl web server used for HTTP debugging (continued)

```
    # (7) prompt for response message, and input response lines,
    #     sending response lines to client, until solitary "."
    printf("    <<<Type Response Followed by '.'>>>\n");

    while ($line = <STDIN>)
    {
        $line =~ s/\r//;
        $line =~ s/\n//;
        if ($line =~ /^\./) { last; }
        print C $line . "\r\n";
    }
    close(C);
}
```

Figure 5-2 shows how the administrator of Joe's Hardware store might use *type-o-serve* to test HTTP communication:

- First, the administrator starts the *type-o-serve* diagnostic server, listening on a particular port. Because Joe's Hardware store already has a production web server listing on port 80, the administrator starts the *type-o-serve* server on port 8080 (you can pick any unused port) with this command line:

  ```
  % type-o-serve.pl 8080
  ```

- Once *type-o-serve* is running, you can point a browser to this web server. In Figure 5-2, we browse to *http://www.joes-hardware.com:8080/foo/bar/blah.txt*.
- The *type-o-serve* program receives the HTTP request message from the browser and prints the contents of the HTTP request message on screen. The *type-o-serve* diagnostic tool then waits for the user to type in a simple response message, followed by a period on a blank line.
- *type-o-serve* sends the HTTP response message back to the browser, and the browser displays the body of the response message.

What Real Web Servers Do

The Perl server we showed in Example 5-1 is a trivial example web server. State-of-the-art commercial web servers are much more complicated, but they do perform several common tasks, as shown in Figure 5-3:

1. Set up connection—accept a client connection, or close if the client is unwanted.
2. Receive request—read an HTTP request message from the network.
3. Process request—interpret the request message and take action.
4. Access resource—access the resource specified in the message.
5. Construct response—create the HTTP response message with the right headers.
6. Send response—send the response back to the client.
7. Log transaction—place notes about the completed transaction in a log file.

Google Exhibit No. 1006
Page 055

Google Exhibit No. 1063
Page 0456

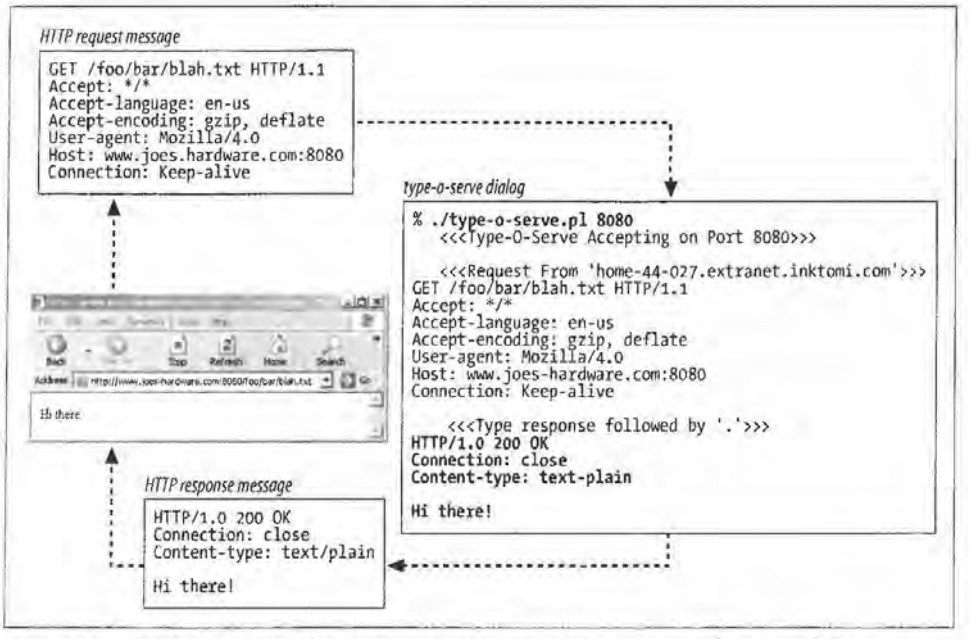


Figure 5-2. The type-o-serve utility lets you type in server responses to send back to clients

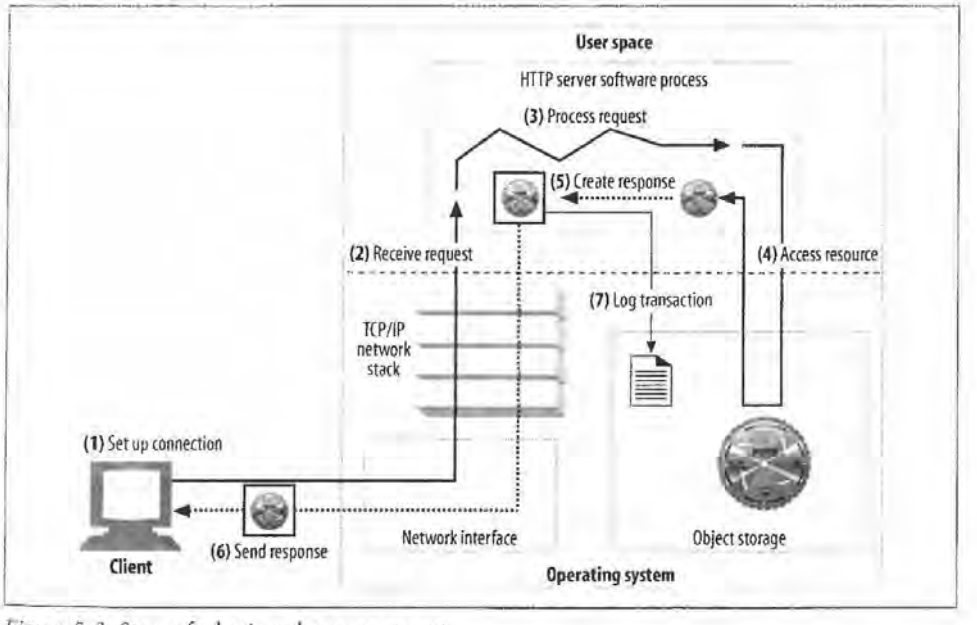


Figure 5-3. Steps of a basic web server request

The next seven sections highlight how web servers perform these basic tasks.

Google Exhibit No. 1006
Page 056

Google Exhibit No. 1063
Page 0457

Step 3: Processing Requests

Once the web server has received a request, it can process the request using the method, resource, headers, and optional body.

Some methods (e.g., POST) require entity body data in the request message. Other methods (e.g., OPTIONS) allow a request body but don't require one. A few methods (e.g., GET) forbid entity body data in request messages.

We won't talk about request processing here, because it's the subject of most of the chapters in the rest of this book!

Step 4: Mapping and Accessing Resources

Web servers are resource servers. They deliver precreated content, such as HTML pages or JPEG images, as well as dynamic content from resource-generating applications running on the servers.

Before the web server can deliver content to the client, it needs to identify the source of the content, by mapping the URI from the request message to the proper content or content generator on the web server.

Docroots

Web servers support different kinds of resource mapping, but the simplest form of resource mapping uses the request URI to name a file in the web server's filesystem. Typically, a special folder in the web server filesystem is reserved for web content. This folder is called the *document root*, or *docroot*. The web server takes the URI from the request message and appends it to the document root.

In Figure 5-8, a request arrives for */specials/saw-blade.gif*. The web server in this example has document root */usr/local/httpd/files*. The web server returns the file */usr/local/httpd/files/specials/saw-blade.gif*.

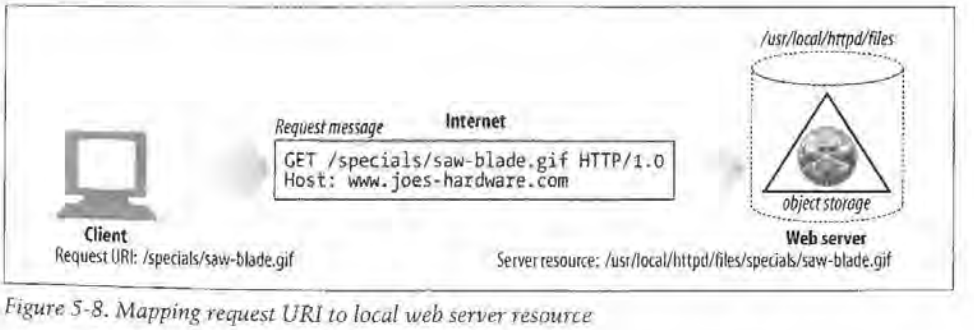


Figure 5-8. Mapping request URI to local web server resource

Google Exhibit No. 1006
Page 057

Google Exhibit No. 1063
Page 0458

To set the document root for an Apache web server, add a *DocumentRoot* line to the *httpd.conf* configuration file:

```
DocumentRoot /usr/local/httpd/files
```

Servers are careful not to let relative URLs back up out of a docroot and expose other parts of the filesystem. For example, most mature web servers will not permit this URI to see files above the Joe's Hardware document root:

http://www.joes-hardware.com/../

Virtually hosted docroots

Virtually hosted web servers host multiple web sites on the same web server, giving each site its own distinct document root on the server. A virtually hosted web server identifies the correct document root to use from the IP address or hostname in the URI or the Host header. This way, two web sites hosted on the same web server can have completely distinct content, even if the request URIs are identical.

In Figure 5-9, the server hosts two sites: *www.joes-hardware.com* and *www.marys-antiques.com*. The server can distinguish the web sites using the HTTP Host header, or from distinct IP addresses.

- When request A arrives, the server fetches the file for */docs/joe/index.html*.
- When request B arrives, the server fetches the file for */docs/mary/index.html*.

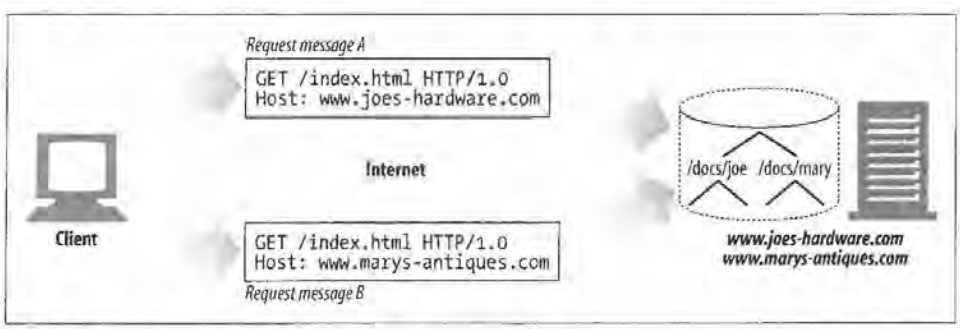


Figure 5-9. Different docroots for virtually hosted requests

Configuring virtually hosted docroots is simple for most web servers. For the popular Apache web server, you need to configure a *VirtualHost* block for each virtual web site, and include the *DocumentRoot* for each virtual server (Example 5-3).

Example 5-3. Apache web server virtual host docroot configuration

```
<VirtualHost www.joes-hardware.com>
  ServerName www.joes-hardware.com
  DocumentRoot /docs/joe
```


Google Exhibit No. 1006
Page 058

Google Exhibit No. 1063
Page 0459

Example 5-3. Apache web server virtual host docroot configuration (continued)

```
  TransferLog /logs/joe.access_log
  ErrorLog /logs/joe.error_log
</VirtualHost>

<VirtualHost www.marys-antiques.com>
  ServerName www.marys-antiques.com
  DocumentRoot /docs/mary
  TransferLog /logs/mary.access_log
  ErrorLog /logs/mary.error_log
</VirtualHost>
  ...
```

Look forward to "Virtual Hosting" in Chapter 18 for much more detail about virtual hosting.

User home directory docroots

Another common use of docroots gives people private web sites on a web server. A typical convention maps URIs whose paths begin with a slash and tilde (/~) followed by a username to a private document root for that user. The private docroot is often the folder called *public_html* inside that user's home directory, but it can be configured differently (Figure 5-10).

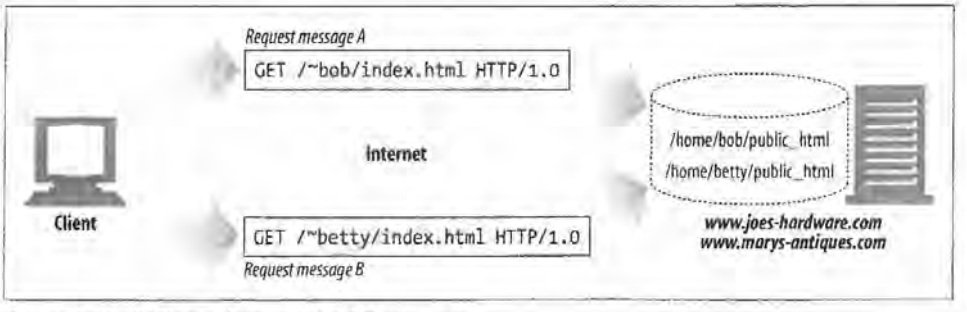


Figure 5-10. Different docroots for different users

Directory Listings

A web server can receive requests for directory URLs, where the path resolves to a directory, not a file. Most web servers can be configured to take a few different actions when a client requests a directory URL:

- Return an error.
- Return a special, default, "index file" instead of the directory.
- Scan the directory, and return an HTML page containing the contents.

Most web servers look for a file named *index.html* or *index.htm* inside a directory to represent that directory. If a user requests a URI. for a directory and the directory

Google Exhibit No. 1006
Page 059

Google Exhibit No. 1063
Page 0460

contains a file named *index.html* (or *index.htm*), the server will return the contents of that file.

In the Apache web server, you can configure the set of filenames that will be interpreted as default directory files using the *DirectoryIndex* configuration directive. The *DirectoryIndex* directive lists all filenames that serve as directory index files, in preferred order. The following configuration line causes Apache to search a directory for any of the listed files in response to a directory URL request:

```
DirectoryIndex index.html index.htm home.html home.htm index.cgi
```

If no default index file is present when a user requests a directory URI, and if directory indexes are not disabled, many web servers automatically return an HTML file listing the files in that directory, and the sizes and modification dates of each file, including URI links to each file. This file listing can be convenient, but it also allows nosy people to find files on a web server that they might not normally find.

You can disable the automatic generation of directory index files with the Apache directive:

```
Options -Indexes
```

Dynamic Content Resource Mapping

Web servers also can map URIs to dynamic resources—that is, to programs that generate content on demand (Figure 5-11). In fact, a whole class of web servers called *application servers* connect web servers to sophisticated backend applications. The web server needs to be able to tell when a resource is a dynamic resource, where the dynamic content generator program is located, and how to run the program. Most web servers provide basic mechanisms to identify and map dynamic resources.

Apache lets you map URI pathname components into executable program directories. When a server receives a request for a URI with an executable path component, it attempts to execute a program in a corresponding server directory. For example, the following Apache configuration directive specifies that all URIs whose paths begin with */cgi-bin/* should execute corresponding programs found in the directory */usr/local/etc/httpd/cgi-programs/*:

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-programs/
```

Apache also lets you mark executable files with a special file extension. This way, executable scripts can be placed in any directory. The following Apache configuration directive specifies that all web resources ending in *.cgi* should be executed:

```
AddHandler cgi-script .cgi
```

CGI is an early, simple, and popular interface for executing server-side applications. Modern application servers have more powerful and efficient server-side dynamic content support, including Microsoft's Active Server Pages and Java servlets.

Google Exhibit No. 1006
Page 060

Google Exhibit No. 1063
Page 0461

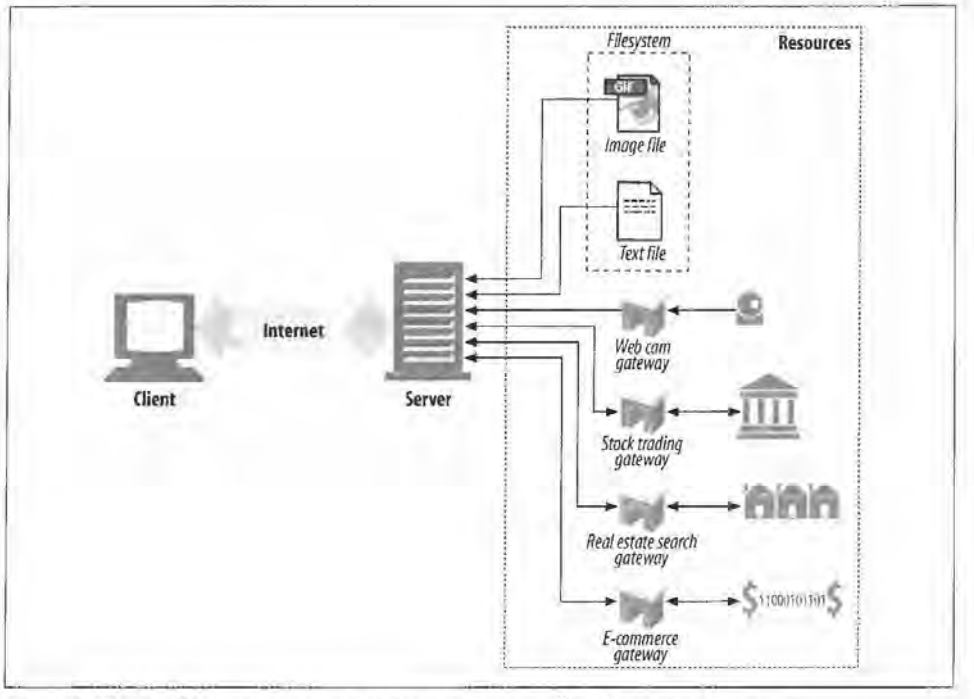


Figure 5-11. A web server can serve static resources as well as dynamic resources

Server-Side Includes (SSI)

Many web servers also provide support for server-side includes. If a resource is flagged as containing server-side includes, the server processes the resource contents before sending them to the client.

The contents are scanned for certain special patterns (often contained inside special HTML comments), which can be variable names or embedded scripts. The special patterns are replaced with the values of variables or the output of executable scripts. This is an easy way to create dynamic content.

Access Controls

Web servers also can assign access controls to particular resources. When a request arrives for an access-controlled resource, the web server can control access based on the IP address of the client, or it can issue a password challenge to get access to the resource.

Refer to Chapter 12 for more information about HTTP authentication.

Google Exhibit No. 1006
Page 061

Google Exhibit No. 1063
Page 0462

Step 5: Building Responses

Once the web server has identified the resource, it performs the action described in the request method and returns the response message. The response message contains a response status code, response headers, and a response body if one was generated. HTTP response codes were detailed in "Status Codes" in Chapter 3.

Response Entities

If the transaction generated a response body, the content is sent back with the response message. If there was a body, the response message usually contains:

- A Content-Type header, describing the MIME type of the response body
- A Content-Length header, describing the size of the response body
- The actual message body content

MIME Typing

The web server is responsible for determining the MIME type of the response body. There are many ways to configure servers to associate MIME types with resources:

mime.types
: The web server can use the extension of the filename to indicate MIME type. The web server scans a file containing MIME types for each extension to compute the MIME type for each resource. This extension-based type association is the most common; it is illustrated in Figure 5-12.

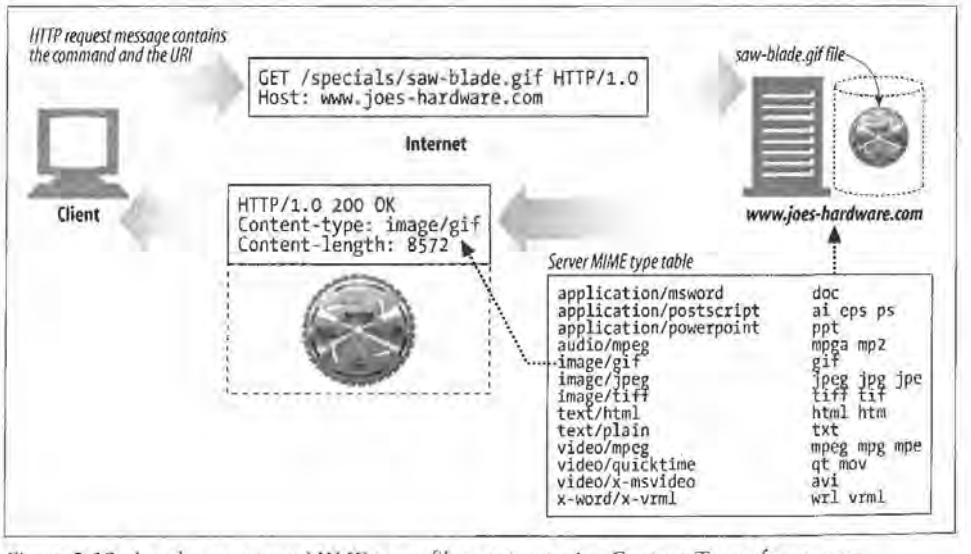


Figure 5-12. A web server uses MIME types file to set outgoing Content-Type of resources

Google Exhibit No. 1006
Page 062

Google Exhibit No. 1063
Page 0463

Magic typing
: The Apache web server can scan the contents of each resource and pattern-match the content against a table of known patterns (called the *magic* file) to determine the MIME type for each file. This can be slow, but it is convenient, especially if the files are named without standard extensions.

Explicit typing
: Web servers can be configured to force particular files or directory contents to have a MIME type, regardless of the file extension or contents.

Type negotiation
: Some web servers can be configured to store a resource in multiple document formats. In this case, the web server can be configured to determine the "best" format to use (and the associated MIME type) by a negotiation process with the user. We'll discuss this in Chapter 17.

Web servers also can be configured to associate particular files with MIME types.

Redirection

Web servers sometimes return redirection responses instead of success messages. A web server can redirect the browser to go elsewhere to perform the request. A redirection response is indicated by a 3XX return code. The Location response header contains a URI for the new or preferred location of the content. Redirects are useful for:

Permanently moved resources
: A resource might have been moved to a new location, or otherwise renamed, giving it a new URL. The web server can tell the client that the resource has been renamed, and the client can update any bookmarks, etc. before fetching the resource from its new location. The status code 301 Moved Permanently is used for this kind of redirect.

Temporarily moved resources
: If a resource is temporarily moved or renamed, the server may want to redirect the client to the new location. But, because the renaming is temporary, the server wants the client to come back with the old URL in the future and not to update any bookmarks. The status codes 303 See Other and 307 Temporary Redirect are used for this kind of redirect.

URL augmentation
: Servers often use redirects to rewrite URLs, often to embed context. When the request arrives, the server generates a new URL containing embedded state information and redirects the user to this new URL.* The client follows the redirect, reissuing the request, but now including the full, state-augmented URL. This is a

* These extended, state-augmented URLs are sometimes called "fat URLs."

Google Exhibit No. 1006
Page 063

Google Exhibit No. 1063
Page 0464

useful way of maintaining state across transactions. The status codes 303 See Other and 307 Temporary Redirect are used for this kind of redirect.

Load balancing
: If an overloaded server gets a request, the server can redirect the client to a less heavily loaded server. The status codes 303 See Other and 307 Temporary Redirect are used for this kind of redirect.

Server affinity
: Web servers may have local information for certain users; a server can redirect the client to a server that contains information about the client. The status codes 303 See Other and 307 Temporary Redirect are used for this kind of redirect.

Canonicalizing directory names
: When a client requests a URI for a directory name without a trailing slash, most web servers redirect the client to a URI with the slash added, so that relative links work correctly.

Step 6: Sending Responses

Web servers face similar issues sending data across connections as they do receiving. The server may have many connections to many clients, some idle, some sending data to the server, and some carrying response data back to the clients.

The server needs to keep track of the connection state and handle persistent connections with special care. For nonpersistent connections, the server is expected to close its side of the connection when the entire message is sent.

For persistent connections, the connection may stay open, in which case the server needs to be extra cautious to compute the Content-Length header correctly, or the client will have no way of knowing when a response ends (see Chapter 4).

Step 7: Logging

Finally, when a transaction is complete, the web server notes an entry into a log file, describing the transaction performed. Most web servers provide several configurable forms of logging. Refer to Chapter 21 for more details.

For More Information

For more information on Apache, Jigsaw, and *ident*, check out:

Apache: The Definitive Guide
: Ben Laurie and Peter Laurie, O'Reilly & Associates, Inc.

Professional Apache
: Peter Wainwright, Wrox Press.

Google Exhibit No. 1006
Page 064

Google Exhibit No. 1063
Page 0465

http://www.w3c.org/Jigsaw/
Jigsaw—W3C's Server W3C Consortium Web Site.

http://www.ietf.org/rfc/rfc1413.txt
RFC 1413, "Identification Protocol," by M. St. Johns.

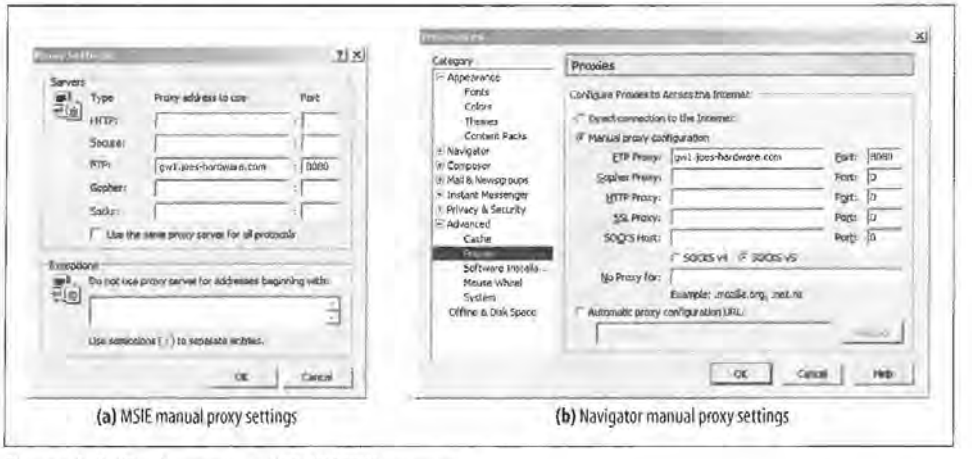
Google Exhibit No. 1006
Page 065

Google Exhibit No. 1063
Page 0466

Protocol Gateways

You can direct HTTP traffic to gateways the same way you direct traffic to proxies. Most commonly, you explicitly configure browsers to use gateways, intercept traffic transparently, or configure gateways as surrogates (reverse proxies).

Figure 8-3 shows the dialog boxes used to configure a browser to use server-side FTP gateways. In the configuration shown, the browser is configured to use *gw1.joes-hardware.com* as an HTTP/FTP gateway for all FTP URLs. Instead of sending FTP commands to an FTP server, the browser will send HTTP commands to the HTTP/FTP gateway *gw1.joes-hardware.com* on port 8080.



(a) MSIE manual proxy settings (b) Navigator manual proxy settings

Figure 8-3. Configuring an HTTP/FTP gateway

The result of this gateway configuration is shown in Figure 8-4. Normal HTTP traffic is unaffected; it continues to flow directly to origin servers. But requests for FTP URLs are sent to the gateway *gw1.joes-hardware.com* within HTTP requests. The gateway performs the FTP transactions on the client's behalf and carries results back to the client by HTTP.

The following sections describe common kinds of gateways: server protocol converters, server-side security gateways, client-side security gateways, and application servers.

HTTP/*: Server-Side Web Gateways

Server-side web gateways convert client-side HTTP requests into a foreign protocol, as the requests travel inbound to the origin server (see Figure 8-5).

In Figure 8-5, the gateway receives an HTTP request for an FTP resource:

```
ftp://ftp.irs.gov/pub/00-index.txt
```


Google Exhibit No. 1006
Page 066

Google Exhibit No. 1063
Page 0467

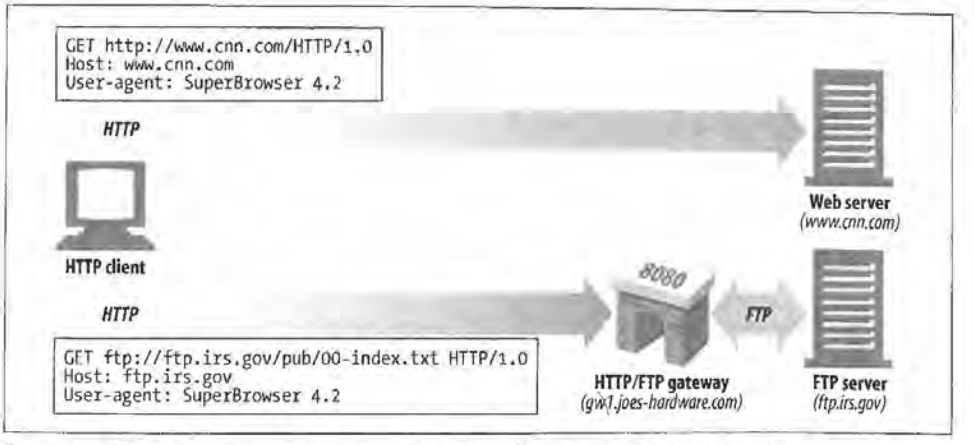


Figure 8-4. Browsers can configure particular protocols to use particular gateways

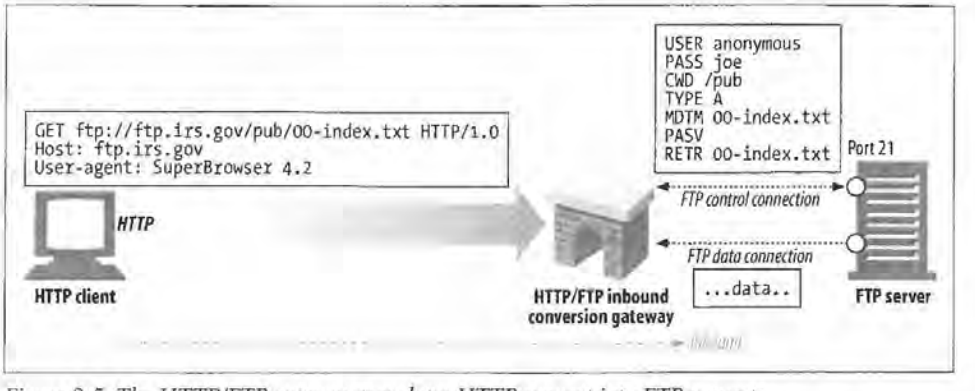


Figure 8-5. The HTTP/FTP gateway translates HTTP request into FTP requests

The gateway proceeds to open an FTP connection to the FTP port on the origin server (port 21) and speak the FTP protocol to fetch the object. The gateway does the following:

- Sends the USER and PASS commands to log in to the server
- Issues the CWD command to change to the proper directory on the server
- Sets the download type to ASCII
- Fetches the document's last modification time with MDTM
- Tells the server to expect a passive data retrieval request using PASV
- Requests the object retrieval using RETR
- Opens a data connection to the FTP server on a port returned on the control channel; as soon as the data channel is opened, the object content flows back to the gateway

Google Exhibit No. 1006
Page 067

Google Exhibit No. 1063
Page 0468

URL is crawled, the corresponding presence bit is set. If the presence bit is already set, the crawler assumes the URL has already been crawled.*

Checkpoints

Be sure to save the list of visited URLs to disk, in case the robot program crashes.

Partitioning

As the Web grows, it may become impractical to complete a crawl with a single robot on a single computer. That computer may not have enough memory, disk space, computing power, or network bandwidth to complete a crawl.

Some large-scale web robots use "farms" of robots, each a separate computer, working in tandem. Each robot is assigned a particular "slice" of URLs, for which it is responsible. Together, the robots work to crawl the Web. The individual robots may need to communicate to pass URLs back and forth, to cover for malfunctioning peers, or to otherwise coordinate their efforts.

A good reference book for implementing huge data structures is *Managing Gigabytes: Compressing and Indexing Documents and Images*, by Witten, et. al (Morgan Kaufmann). This book is full of tricks and techniques for managing large amounts of data.

Aliases and Robot Cycles

Even with the right data structures, it is sometimes difficult to tell if you have visited a page before, because of URL "aliasing." Two URLs are *aliases* if the URLs look different but really refer to the same resource.

Table 9-1 illustrates a few simple ways that different URLs can point to the same resource.

Table 9-1. Different URLs that alias to the same documents

	First URL	Second URL	When aliased
a	*http://www.foo.com/bar.html*	*http://www.foo.com:80/bar.html*	Port is 80 by default
b	*http://www.foo.com/~fred*	*http://www.foo.com/%7Ffred*	%7F is same as ~
c	*http://www.foo.com/x.html#early*	*http://www.foo.com/x.html#middle*	Tags don't change the page
d	*http://www.foo.com/readme.htm*	*http://www.foo.com/README.HTM*	Case-insensitive server
e	*http://www.foo.com/*	*http://www.foo.com/index.html*	Default page is *index.html*
f	*http://www.foo.com/index.html*	*http://209.231.87.45/index.html*	*www.foo.com* has this IP address

* Because there are a potentially infinite number of URLs and only a finite number of bits in the presence bit array, there is potential for collision—two URLs can map to the same presence bit. When this happens, the crawler mistakenly concludes that a page has been crawled when it hasn't. In practice, this situation can be made very unlikely by using a large number of presence bits. The penalty for collision is that a page will be omitted from a crawl.

Google Exhibit No. 1006
Page 068

Google Exhibit No. 1063
Page 0469

Canonicalizing URLs

Most web robots try to eliminate the obvious aliases up front by "canonicalizing" URLs into a standard form. A robot might first convert every URL into a canonical form, by:

1. Adding ":80" to the hostname, if the port isn't specified
2. Converting all %xx escaped characters into their character equivalents
3. Removing # tags

These steps can eliminate the aliasing problems shown in Table 9-1a–c. But, without knowing information about the particular web server, the robot doesn't have any good way of avoiding the duplicates from Table 9-1d–f:

- The robot would need to know whether the web server was case-insensitive to avoid the alias in Table 9-1d.
- The robot would need to know the web server's index-page configuration for this directory to know whether the URLs in Table 9-1e were aliases.
- The robot would need to know if the web server was configured to do virtual hosting (covered in Chapter 5) to know if the URLs in Table 9-1f were aliases, even if it knew the hostname and IP address referred to the same physical computer.

URL canonicalization can eliminate the basic syntactic aliases, but robots will encounter other URL aliases that can't be eliminated through converting URLs to standard forms.

Filesystem Link Cycles

Symbolic links on a filesystem can cause a particularly insidious kind of cycle, because they can create an illusion of an infinitely deep directory hierarchy where none exists. Symbolic link cycles usually are the result of an unintentional error by the server administrator, but they also can be created by "evil webmasters" as a malicious trap for robots.

Figure 9-3 shows two filesystems. In Figure 9-3a, *subdir* is a normal directory. In Figure 9-3b, *subdir* is a symbolic link pointing back to /. In both figures, assume the file */index.html* contains a hyperlink to the file *subdir/index.html*.

Using Figure 9-3a's filesystem, a web crawler may take the following actions:

1. `GET http://www.foo.com/index.html`
   Get */index.html*, find link to *subdir/index.html*.
2. `GET http://www.foo.com/subdir/index.html`
   Get *subdir/index.html*, find link to *subdir/logo.gif*.
3. `GET http://www.foo.com/subdir/logo.gif`
   Get *subdir/logo.gif*, no more links, all done.

Google Exhibit No. 1006
Page 069

Google Exhibit No. 1063
Page 0470

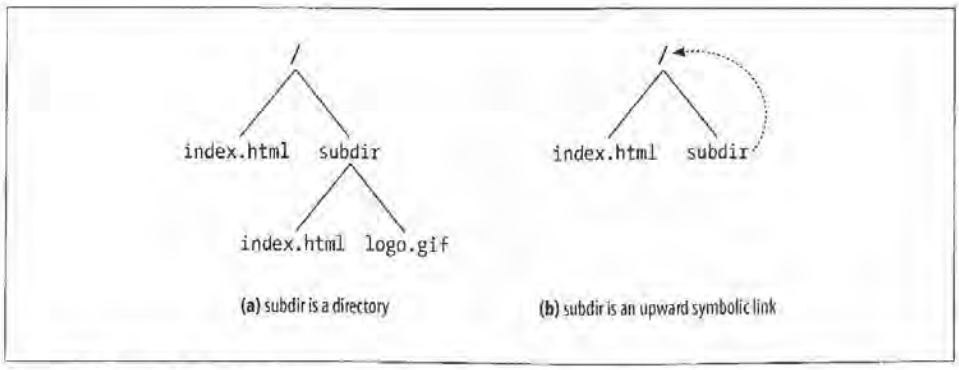


Figure 9-3. Symbolic link cycles

But in Figure 9-3b's filesystem, the following might happen:

1. `GET http://www.foo.com/index.html`
   Get */index.html*, find link to *subdir/index.html*.
2. `GET http://www.foo.com/subdir/index.html`
   Get *subdir/index.html*, but get back same *index.html*.
3. `GET http://www.foo.com/subdir/subdir/index.html`
   Get *subdir/subdir/index.html*.
4. `GET http://www.foo.com/subdir/subdir/subdir/index.html`
   Get *subdir/subdir/subdir/index.html*.

The problem with Figure 9-3b is that *subdir/* is a cycle back to /, but because the URLs look different, the robot doesn't know from the URL alone that the documents are identical. The unsuspecting robot runs the risk of getting into a loop. Without some kind of loop detection, this cycle will continue, often until the length of the URL exceeds the robot's or the server's limits.

Dynamic Virtual Web Spaces

It's possible for malicious webmasters to intentionally create sophisticated crawler loops to trap innocent, unsuspecting robots. In particular, it's easy to publish a URL that looks like a normal file but really is a gateway application. This application can whip up HTML on the fly that contains links to imaginary URLs on the same server. When these imaginary URLs are requested, the nasty server fabricates a new HTML page with new imaginary URLs.

The malicious web server can take the poor robot on an Alice-in-Wonderland journey through an infinite virtual web space, even if the web server doesn't really contain any files. Even worse, it can make it very difficult for the robot to detect the cycle, because the URLs and HTML can look very different each time. Figure 9-4 shows an example of a malicious web server generating bogus content.

Google Exhibit No. 1006
Page 070

Google Exhibit No. 1063
Page 0471

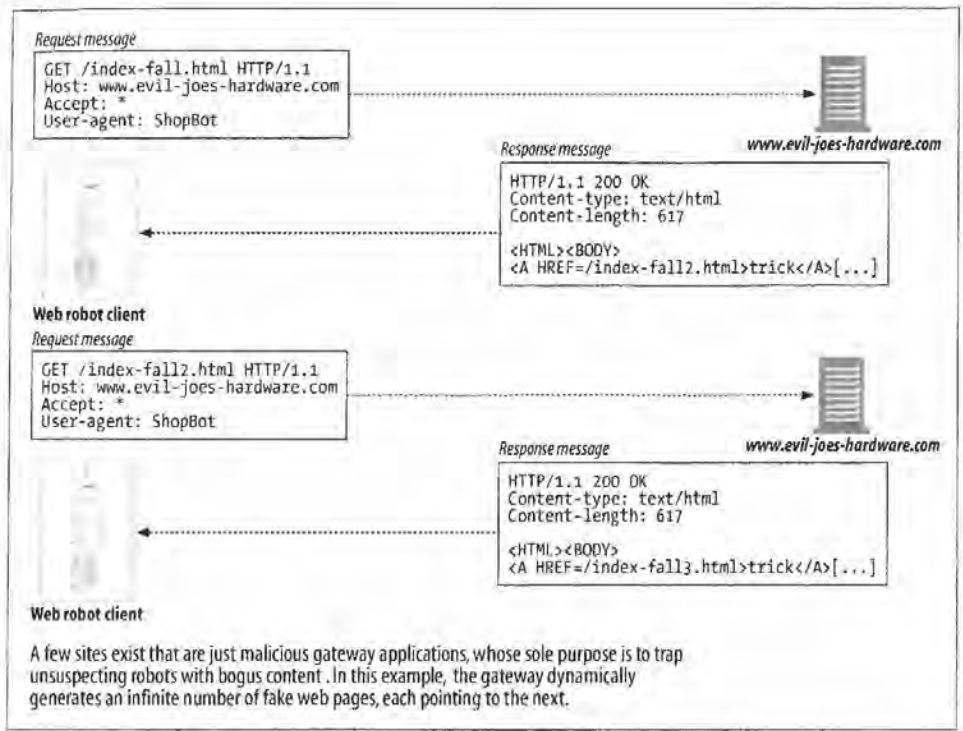


Figure 9-4. Malicious dynamic web space example

More commonly, well-intentioned webmasters may unwittingly create a crawler trap through symbolic links or dynamic content. For example, consider a CGI-based calendaring program that generates a monthly calendar and a link to the next month. A real user would not keep requesting the next-month link forever, but a robot that is unaware of the dynamic nature of the content might keep requesting these resources indefinitely.*

Avoiding Loops and Dups

There is no foolproof way to avoid all cycles. In practice, well-designed robots need to include a set of heuristics to try to avoid cycles.

Generally, the more autonomous a crawler is (less human oversight), the more likely it is to get into trouble. There is a bit of a trade-off that robot implementors need to make—these heuristics can help avoid problems, but they also are somewhat "lossy," because you can end up skipping valid content that looks suspect.

* This is a real example mentioned on *http://www.searchtools.com/robots/robot-checklist.html* for the calendaring site at *http://cgi.umbc.edu/cgi-bin/WebEvent/webevent.cgi*. As a result of dynamic content like this, many robots refuse to crawl pages that have the substring "cgi" anywhere in the URL.

Google Exhibit No. 1006
Page 071

Google Exhibit No. 1063
Page 0472

*Nosy robots**

Some robots may get URLs that point to private data and make that data easily accessible through Internet search engines and other applications. If the owner of the data didn't actively advertise the web pages, she may view the robotic publishing as a nuisance at best and an invasion of privacy at worst.

Usually this happens because a hyperlink to the "private" content that the robot followed already exists (i.e., the content isn't as secret as the owner thought it was, or the owner forgot to remove a preexisting hyperlink). Occasionally it happens when a robot is very zealous in trying to scavenge the documents on a site, perhaps by fetching the contents of a directory, even if no explicit hyperlink exists.

Robot implementors retrieving large amounts of data from the Web should be aware that their robots are likely to retrieve sensitive data at some point—data that the site implementor never intended to be accessible over the Internet. This sensitive data can include password files or even credit card information. Clearly, a mechanism to disregard content once this is pointed out (and remove it from any search index or archive) is important. Malicious search engine and archive users have been known to exploit the abilities of large-scale web crawlers to find content—some search engines, such as Google,† actually archive representations of the pages they have crawled, so even if content is removed, it can still be found and accessed for some time.

Dynamic gateway access

Robots don't always know what they are accessing. A robot may fetch a URL whose content comes from a gateway application. In this case, the data obtained may be special-purpose and may be expensive to compute. Many web site administrators don't like naïve robots requesting documents that come from gateways.

Excluding Robots

The robot community understood the problems that robotic web site access could cause. In 1994, a simple, voluntary technique was proposed to keep robots out of where they don't belong and provide webmasters with a mechanism to better control their behavior. The standard was named the "Robots Exclusion Standard" but is often just called *robots.txt*, after the file where the access-control information is stored.

The idea of *robots.txt* is simple. Any web server can provide an optional file named *robots.txt* in the document root of the server. This file contains information about what robots can access what parts of the server. If a robot follows this voluntary

* Generally, if a resource is available over the public Internet, it is likely referenced somewhere. Few resources are truly private, with the web of links that exists on the Internet.

† See search results at *http://www.google.com*. A cached link, which is a copy of the page that the Google crawler retrieved and indexed, is available on most results.

Google Exhibit No. 1006
Page 072

Google Exhibit No. 1063
Page 0473

```
        <a:ishidden b:dt="boolean">0</a:ishidden>
        <a:iscollection b:dt="boolean">0</a:iscollection>
      </a:prop>
    </a:propstat>
  </a:response></a:multistatus>
```

In this example, the server responds with a 207 Multi-Status code. WebDAV uses the 207 response for PROPFIND and a few other WebDAV methods that act simultaneously on multiple resources and potentially have different responses for each resource.

A few XML elements in the response need to be defined:

<multistatus>
: A container for multiple responses.

<href>
: Identifies the resource's URI.

<status>
: Contains the HTTP status code for the particular request.

<propstat>
: Groups one <status> element and one <prop> element. The <prop> element may contain one or more property name/value pairs for the given resource.

In the sample response listed above, the response is for one URI, *http://minstar/ch-publish.fm*. The <propstar> element embeds one <status> element and one <prop> element. For this URI, the server returned a 200 OK response, as defined by the <status> element. The <prop> element has several subelements; only some are listed in the example.

One instant application of PROPFIND is the support for directory listing. Given the expressability of a PROPFIND request, one single call can retrieve the entire hierarchy of the collection with all the properties of individual entities.

The PROPPATCH Method

The PROPPATCH method provides an atomic mechanism to set or remove multiple properties on a given resource. The atomicity will guarantee that either all of the requests are successful or none of them made it.

The base XML element for the PROPPATCH method is <propertyupdate>. It acts as a container for all the properties that need updating. The XML elements <set> and <remove> are used to specify the operation:

<set>
: Specifies the property values to be set. The <set> contains one or more <prop> subelements, which in turn contains the name/value pairs of the properties to be set for the resource. If the property already exists, the value is replaced.

Google Exhibit No. 1006
Page 073

Google Exhibit No. 1063
Page 0474

<remove>

Specifies the properties that are to be removed. Unlike with <set>, only the names of the properties are listed in the <prop> container.

This trivial example sets and removes the "owner" property:

```
<d:propertyupdate xmlns:d="DAV:" xmlns:o="http://name-space/scheme/">
 <d:set>
   <d:prop>
     <o:owner>Author A</o:owner>
   </d:prop>
 </d:set>

 <d:remove>
   <d:prop>
     <o:owner/>
   </d:prop>
 </d:remove>
</d:propertyupdate>
```

The response to PROPPATCH requests is very similar to that for PROPFIND requests. For more information, refer to RFC 2518.

Table 19-2 summarizes the status codes for the PROPFIND and PROPPATCH methods.

Table 19-2. Status codes for PROPFIND and PROPPATCH methods

Status code	Defined by	Methods	Effect
200 OK	HTTP	PROPFIND, PROPPATCH	Command success.
207 Multi-Status	WEBDAV	PROPFIND, PROPPATCH	When acting on one or more resources (or a collection), the status for each object is encapsulated into one 207 response. This is a typical success response.
401 Unauthorized	HTTP	PROPATCH	Requires authorization to complete the property modification operation.
403 Forbidden	HTTP	PROPFIND, PROPPATCH	For PROPFIND, the client is not allowed to access the property. For PROPPATCH, the client may not change the property.
404 Not Found	HTTP	PROPFIND	No such property.
409 Conflict	HTTP	PROPPATCH	Conflict of update semantics—for example, trying to update a read-only property.
423 Locked	WebDAV	PROPPATCH	Destination resource is locked and there is no lock token or the lock token does not match.
507 Insufficient Storage	WebDAV	PROPPATCH	Not enough space for registering the modified property.

Collections and Namespace Management

A collection refers to a logical or physical grouping of resources in a predefined hierachy. A classic example of a collection is a directory. Like directories in a filesystem,

Google Exhibit No. 1006
Page 074

Google Exhibit No. 1063
Page 0475

collections act as containers of other resources, including other collections (equivalent to directories on the filesystem).

WebDAV uses the XML namespace mechanism. Unlike traditional namespaces, XML namespace partitions allow for precise structural control while preventing any namespace collisions.

WebDAV provides five methods for manipulating the namespace: DELETE, MKCOL, COPY, MOVE, and PROPFIND. PROPFIND was discussed previously in this chapter, but let's talk about the other methods.

The MKCOL Method

The MKCOL method allows clients to create a collection at the indicated URL on the server. At first sight, it may seem rather redundant to define an entire new method just for creating a collection. Overlaying on top of a PUT or POST method seems like a perfect alternative. The designers of the WebDAV protocol did consider these alternatives and still chose to define a new method. Some of the reasons behind that decision are:

- To have a PUT or a POST create a collection, the client needs to send some extra "semantic glue" along with the request. While this certainly is feasible, defining an ad hoc protocol may become tedious and error-prone.
- Most of the access-control mechanisms are based on the type of methods—only a few are allowed to create and delete resources in the repository. If we overload other methods, these access-control mechanisms will not work.

For example, a request might be:

```
MKCOL /publishing HTTP/1.1
Host: minstar
Content-Length: 0
Connection: Keep-Alive
```

And the response might be:

```
HTTP/1.1 201 Created
Server: Microsoft-IIS/5.0
Date: Fri, 10 May 2002 23:20:36 GMT
Location: http://minstar/publishing/
Content-Length: 0
```

Let us examine a few pathological cases:

- Suppose the collection already exists. If a MKCOL /colA request is made and colA already exists (i.e., namespace conflict), the request will fail with a 405 Method Not Allowed status code.
- If there are no write permissions, the MKCOL request will fail with a 403 Forbidden status code.

Google Exhibit No. 1006
Page 075

Google Exhibit No. 1063
Page 0476

- If a request such as MKCOL /colA/colB is made and colA does not exist, the request will fail with a 409 Conflict status code.

Once the file or collection is created, we can delete it with the DELETE method.

The DELETE Method

We already saw the DELETE method in Chapter 3. WebDAV extends the semantics to cover collections.

If we need to delete a directory, the Depth header is needed. If the Depth header is not specified, the DELETE method assumes the Depth header to be set to infinity—that is, all the files in the directory and any subdirectories thereof are deleted. The response also has a Content-Location header identifying the collection that just got deleted. The request might read:

```
DELETE /publishing HTTP/1.0
Host: minstar
```

And the response might read:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Tue, 14 May 2002 16:41:44 GMT
Content-Location: http://minstar/publishing/
Content-Type: text/xml
Content-Length: 0
```

When removing collections, there always is a chance that a file in the collection is locked by someone else and can't be deleted. In such a case, the collection itself can't be deleted, and the server replies with a 207 Multi-Status status code. The request might read:

```
DELETE /publishing HTTP/1.0
Host: minstar
```

And the response might read:

```
HTTP/1.1 207 Multi-Status
Server: Microsoft-IIS/5.0
Content-Location: http://minstar/publishing/
.............
<?xml version="1.0"?>
<a:multistatus xmlns:a="DAV:">
<a:response>
<a:href>http://minstar/index3/ch-publish.fm</a:href>
<a:status> HTTP/1.1 423 Locked </a:status>
</a:response>
</a:multistatus>
```

In this transaction, the <status> XML element contains the status code 423 Locked, indicating that the resource *ch-publish.fm* is locked by another user.

Google Exhibit No. 1006
Page 076

Google Exhibit No. 1063
Page 0477

Table D-3. "Application" MIME types (continued)

MIME type	Description	Extension	Contact and reference
application/x-ustar	The extended tar interchange format.	ustar	See the IEEE 1003.1(1990) specifications
application/x-wais-source	WAIS source structure.	src	
application/xml	Extensible Markup Language format file (use text/xml if you want the file treated as plain text by browsers, etc.).	xml, dtd	RFC 2376
application/zip	PKWARE zip archives.	zip	

audio/*

Table D-4 summarizes audio content types.

Table D-4. "Audio" MIME types

MIME type	Description	Extension	Contact and reference
audio/32kadpcm	8 kHz ADPCM audio encoding.		RFC 2421
audio/basic	Audio encoded with 8-kHz monaural 8-bit ISDN u-law PCM.	au, snd	RFC 1341
audio/G.772.1	G.722.1 compresses 50Hz–7kHz audio signals into 24 kbit/s or 32 kbit/s. It may be used for speech, music, and other types of audio.		RFC 3047
audio/L16	Audio/L16 is based on L16, described in RFC 1890. L16 denotes uncompressed audio data, using 16-bit signed representation.		RFC 2586
audio/MP4A-LATM	MPEG-4 audio.		RFC 3016
audio/midi	MIDI music files.	mid, midi, kar	
audio/mpeg	MPEG encoded audio files.	mpga, mp2, mp3	RFC 3003
audio/parityfec	Parity-based forward error correction for RTP audio.		RFC 3009
audio/prs.sid	Commodore 64 SID audio files.	sid, psid	*http://www.geocities.com/SiliconValley/Lakes/5147/sidplay/docs.html#fileformats*
audio/telephone-event	Logical telephone event.		RFC 2833
audio/tone	Telephonic sound pattern.		RFC 2833
audio/vnd.cns.anp1	Supports voice and unified messaging application features available on the Access NP network services platform from Comverse Network Systems.		Ann McLaughlin Comverse Network Systems *amclaughlin@comversens.com*

Google Exhibit No. 1006
Page 077

Google Exhibit No. 1063
Page 0478

Table D-4. "Audio" MIME types (continued)

MIME type	Description	Extension	Contact and reference
audio/vnd.cns.inf1	Supports voice and unified messaging application features available on the TRILOGUE Infinity network services platform from Comverse Network Systems.		Same as above
audio/vnd.digital-winds	Digital Winds music is never-ending, reproducible, and interactive MIDI music in very small packages (<3K).	eol	Armands Strazds *armands.strazds@medienhaus-bremen.de*
audio/vnd.everad.plj	Proprietary EverAD audio encoding.	plj	Tomer Weisberg *tomer@everad.com*
audio/vnd.lucent.voice	Voice messaging including Lucent Technologies' Intuity™ AUDIX® Multimedia Messaging System and the Lucent Voice Player.	lvp	Frederick Block *rickblock@lucent.com http://www.lucent.com/lvp/*
audio/vnd.nortel.vbk	Proprietary Nortel Networks Voice Block audio encoding.	vbk	Glenn Parsons *Glenn.Parsons@NortelNetworks.com*
audio/vnd.nuera.ecelp4800	Proprietary Nuera Communications audio and speech encoding, available in Nuera voice-over-IP gateways, terminals, application servers, and as a media service for various host platforms and OSs.	ecelp4800	Michael Fox *mfox@nuera.com*
audio/vnd.nuera.ecelp7470	Same as above.	ecelp7470	Same as above
audio/vnd.nuera.ecelp9600	Same as above.	ecelp9600	Same as above
audio/vnd.octel.sbc	Variable-rate encoding averaging 18 kbps used for voice messaging in Lucent Technologies' Sierra™, Overture™, and IMA™ platforms.		Jeff Bouis *jbouis@lucent.com*
audio/vnd.qcelp	Qualcomm audio encoding.	qcp	Andy Dejaco *adejaco@qualcomm.com*
audio/vnd.rhetorex.32kadpcm	32-kbps Rhetorex™ ADPCM audio encoding used in voice messaging products such as Lucent Technologies's CallPerformer™, Unified Messenger™, and other products.		Jeff Bouis *jbouis@lucent.com*
audio/vnd.vmx.cvsd	Audio encoding used in voice messaging products including Lucent Technologies' Overture200™, Overture 300™, and VMX 300™ product lines.		Same as above
audio/x-aiff	AIFF audio file format.	aif, aiff, aifc	
audio/x-pn-realaudio	RealAudio metafile format by Real Networks (formerly Progressive Networks).	ram, rm	
audio/x-pn-realaudio-plugin	From Apache *mime.types*.	rpm	

Google Exhibit No. 1006
Page 078

Google Exhibit No. 1063
Page 0479

Table D-4. "Audio" MIME types (continued)

MIME type	Description	Extension	Contact and reference
audio/x-realaudio	RealAudio audio format by Real Networks (formerly Progressive Networks).	ra	
audio/x-wav	WAV audio files.	wav	

chemical/*

Much of the information in Table D-5 was obtained courtesy of the "Chemical MIME Home Page" (*http://www.ch.ic.ac.uk/chemime/*).

Table D-5. "Chemical" MIME types

MIME type	Description	Extension	Contact and reference
chemical/x-alchemy	Alchemy format	alc	*http://www.camsoft.com*
chemical/x-cache-csf		csf	
chemical/x-cactvs-binary	CACTVS binary format	cbin	*http://cactvs.cit.nih.gov*
chemical/x-cactvs-ascii	CACTVS ASCII format	cascii	*http://cactvs.cit.nih.gov*
chemical/x-cactvs-table	CACTVS table format	ctab	*http://cactvs.cit.nih.gov*
chemical/x-cdx	ChemDraw eXchange file	cdx	*http://www.camsoft.com*
chemical/x-cerius	MSI Cerius II format	cer	*http://www.msi.com*
chemical/x-chemdraw	ChemDraw file	chm	*http://www.camsoft.com*
chemical/x-cif	Crystallographic Interchange Format	cif	*http://www.bernstein-plus-sons.com/software/rasmol/* *http://ndbserver.rutgers.edu/NDB/mmcif/examples/index.html*
chemical/x-mmcif	MacroMolecular CIF	mcif	Same as above
chemical/x-chem3d	Chem3D format	c3d	*http://www.camsoft.com*
chemical/x-cmdf	CrystalMaker Data Format	cmdf	*http://www.crystalmaker.co.uk*
chemical/x-compass	Compass program of the Takahashi	cpa	
chemical/x-crossfire	Crossfire file	bsd	
chemical/x-cml	Chemical Markup Language	cml	*http://www.xml-cml.org*
chemical/x-csml	Chemical Style Markup Language	csml, csm	*http://www.mdli.com*
chemical/x-ctx	Gasteiger group CTX file format	ctx	
chemical/x-cxf		cxf	
chemical/x-daylight-smiles	Smiles format	smi	*http://www.daylight.com/dayhtml/smiles/index.html*
chemical/x-embl-dl-nucleotide	EMBL nucleotide format	emb	*http://mercury.ebi.ac.uk*
chemical/x-galactic-spc	SPC format for spectral and chromatographic data	spc	*http://www.galactic.com/galactic/Data/spcvue.htm*

Google Exhibit No. 1006
Page 079

Google Exhibit No. 1063
Page 0480

Table D-10. "Text" MIME types (continued)

MIME type	Description	Extension	Contact and reference
text/vnd.wap.si	Service Indication (SI) objects contain a message describing an event and a URI describing where to load the corresponding service.	si, xml	WAP Forum Ltd *http://www.wapforum.org*
text/vnd.wap.sl	The Service Loading (SL) content type provides a means to convey a URI to a user agent in a mobile client. The client itself automatically loads the content indicated by that URI and executes it in the addressed user agent without user intervention when appropriate.	sl, xml	Same as above
text/vnd.wap.wml	Wireless Markup Language (WML) is a markup language, based on XML, that defines content and user interface for narrow-band devices, including cellular phones and pagers.	wml	Same as above
text/vnd.wap.wmlscript	WMLScript is an evolution of JavaScript for wireless devices.	wmls	Same as above
text/x-setext	From Apache *mime.types*.	etx	
text/xml	Extensible Markup Language format file (use application/xml if you want the browser to save to file when downloaded).	xml	RFC 2376

video/*

Table D-11 lists some popular video movie formats. Note that some video formats are classified as application types.

Table D-11. "Video" MIME types

MIME type	Description	Extension	Contact and reference
video/MP4V-ES	MPEG-4 video payload, as carried by RTP.		RFC 3016
video/mpeg	Video encoded per the ISO 11172 CD MPEG standard.	mpeg, mpg, mpe	RFC 1341
video/parityfec	Forward error correcting video format for data carried through RTP streams.		RFC 3009
video/pointer	Transporting pointer position information for presentations.		RFC 2862
video/quicktime	Apple Quicktime video format.	qt, mov	*http://www.apple.com*
video/vnd.fvt	Video format from FAST Search & Transfer.	fvt	Arild Fuldseth *Arild.Fuldseth@fast.no*

Google Exhibit No. 1006
Page 080

Google Exhibit No. 1063
Page 0481

Table D-11. "Video" MIME types (continued)

MIME type	Description	Extension	Contact and reference
video/vnd.motorola.video video/vnd.motorola.videop	Proprietary formats used by products from Motorola ISG.		Tom McGinty Motorola ISG *tmcginty@dma.isg.mot*
video/vnd.mpegurl	This media type consists of a series of URLs of MPEG Video files.	mxu	Heiko Recktenwald *uzs106@uni-bonn.de* "Power and Responsibility: Conversations with Contributors," Guy van Belle, et al., LMJ 9 (1999), 127–133, 129 (MIT Press)
video/vnd.nokia.interleaved-multimedia	Used in Nokia 9210 Communicator video player and related tools.	nim	Petteri Kangaslampi *petteri.kangaslampi@nokia.com*
video/x-msvideo	Microsoft AVI movies.	avi	*http://www.microsoft.com*
video/x-sgi-movie	Silicon Graphics's movie format.	movie	*http://www.sgi.com*

Experimental Types

The set of primary types supports most content types. Table D-12 lists one experimental type, for conferencing software, that is configured in some web servers.

Table D-12. Extension MIME types

MIME type	Description	Extension	Contact and reference
x-conference/x-cooltalk	Collaboration tool from Netscape	ice	

Google Exhibit No. 1006
Page 081

Google Exhibit No. 1063
Page 0482


HTTP
The Definitive Guide
Gourley, Totty, Sayer, Reddy & Aggarwal
O'REILLY

Web Programming

O'REILLY®

HTTP: The Definitive Guide



Behind every successful web transaction lurks the Hypertext Transfer Protocol (HTTP), the language by which web clients and servers exchange documents and information. HTTP is commonly known as the workhorse behind the browsers we use every day to access our company intranets, locate out-of-print books, or research census information. But HTTP is used for far more than browsing the Web: the simplicity and ubiquity of HTTP also have made it the choice protocol for many other networked applications, most notably through web services such as SOAP and XML-RPC.

As the title suggests, *HTTP: The Definitive Guide* explains the HTTP protocol: how it works and how to use it to develop web-based applications. However, this book is not just about HTTP; it's also about all the other core Internet technologies that HTTP depends on to work effectively. Although HTTP is at the center of the book, the essence of *HTTP: The Definitive Guide* is in understanding how the Web works and how to apply that knowledge to web programming and administration. The book explains the technical workings, motivations, performance considerations, and objectives of HTTP and the technologies around which it revolves.

HTTP: The Definitive Guide is the bible for the HTTP protocol and related web technologies. Topics covered include:

- HTTP methods, headers, and status codes
- Optimizing proxies and caches
- Strategies for designing web robots and crawlers
- Cookies, authentication, and Secure HTTP
- Internationalization and content negotiation
- Redirection and load-balancing strategies

Written by experts with years of practical experience, this book uses clear, concise language and a plethora of detailed illustrations to help readers visualize what goes on behind the scenes, providing a complete understanding of the story behind each query on the Web.

All web programmers, administrators, and application developers need to be familiar with HTTP in order to work effectively. There are many books that explain how to use the Web, but this is the book that explains how the Web works.

www.oreilly.com

US $54.99 **CAN $63.99**

ISBN: 978-1-56592-509-0

9 781565 925090 55499

Ex. 1006A

Ex. 1006B

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 = HTTP The definitive guide
Search Results: Displaying 1 of 7 entries

previous next

Labeled View

HTTP : the definitive guide / David Gourley and Brian Totty, with Marjorie...

Type of Work: Text
Registration Number / Date: TX0005747823 / 2002-10-23
Title: HTTP : the definitive guide / David Gourley and Brian Totty, with Marjorie Sayer, Sailu Reddy, and Anshu Aggarwal.
Imprint: Beijing ; Sebastopol, CA : O'Reilly, c2002.
Description: 635 p.
Copyright Claimant: O'Reilly & Associates, Inc.
Date of Creation: 2002
Date of Publication: 2002-09-19
Copyright Note: C.O. correspondence.
Variant title: HTTP : the definitive guide
Names: Gourley, David
Totty, Brian
Sayer, Marjorie
Reddy, Sailu
Aggarwal, Anshu
O'Reilly & Associates, Inc.

previous next

Save, Print and Email (Help Page)

Select Download Format: Full Record | Format for Print/Save

Enter your email address: | Email

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

Google Exhibit No. 1063
Page 0487

ICICS-PCM 2003
15-18 December 2003
Singapore

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

Shih-Hao Wang, Wen-Hsiao Peng, Yuwen He[1], 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
[1]Department 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

0-7803-8185-8/03/$17.00 © 2003 IEEE

Google Exhibit No. 1007
Page 001

Google Exhibit No. 1063
Page 0488

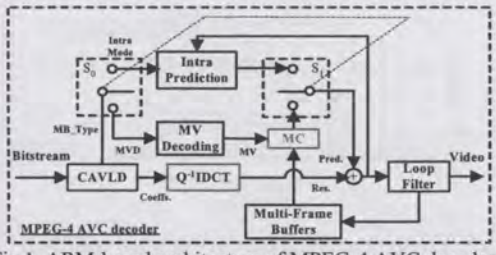


Fig.1. ARM-based architecture of MPEG-4 AVC decoder

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

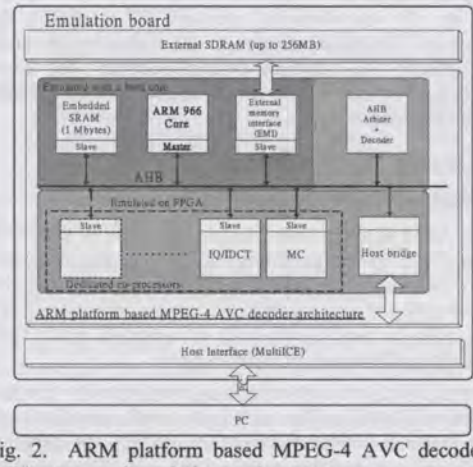


Fig. 2. ARM platform based MPEG-4 AVC decoder architecture and emulation environment.

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

Google Exhibit No. 1007
Page 002

Google Exhibit No. 1063
Page 0489

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,

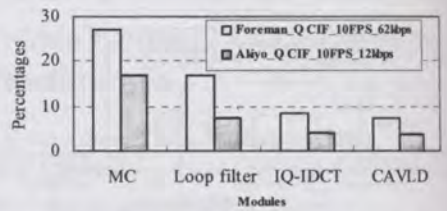


Fig. 3. Decoding profiling for AVC on ARM 966 CPU.

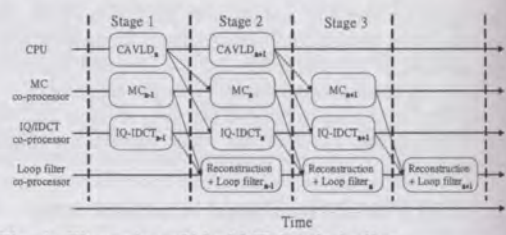


Fig. 4. Illustration of ideal MB level pipeline.

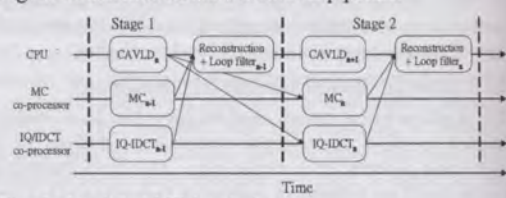


Fig. 5. Modified MB level pipeline.

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.

Google Exhibit No. 1007
Page 003

Google Exhibit No. 1063
Page 0490

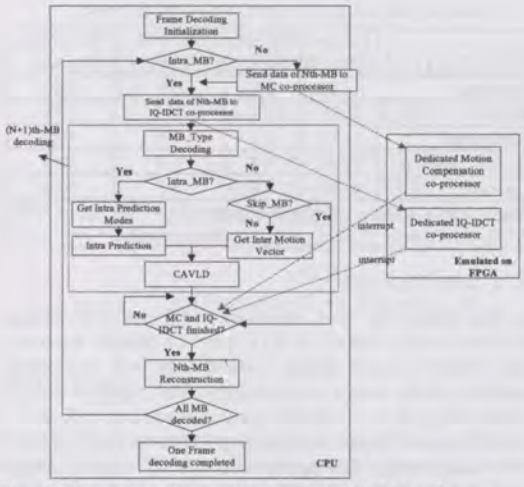


Fig. 6. Flow chart for the MB level pipeline in software.

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.

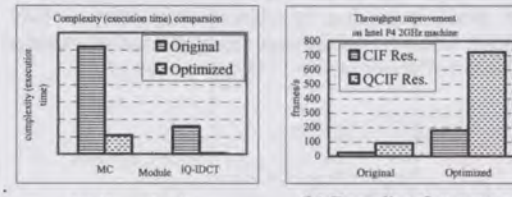
C. Content aware IQ-IDCT

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.

	Fast motion	Slow motion
Sequence(QCIF)	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,

Google Exhibit No. 1007
Page 004

Google Exhibit No. 1063
Page 0491

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.

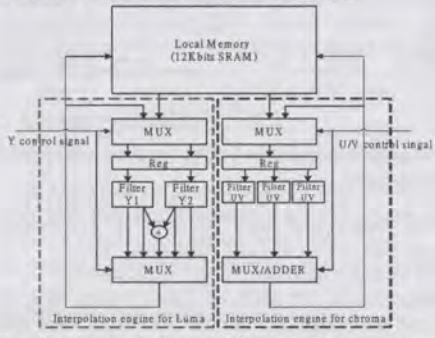


Fig. 8. Interpolation architecture for quarter-pel motion compensation

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 fps [1] 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.

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

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= 1I + 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*, "Muiticore 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.

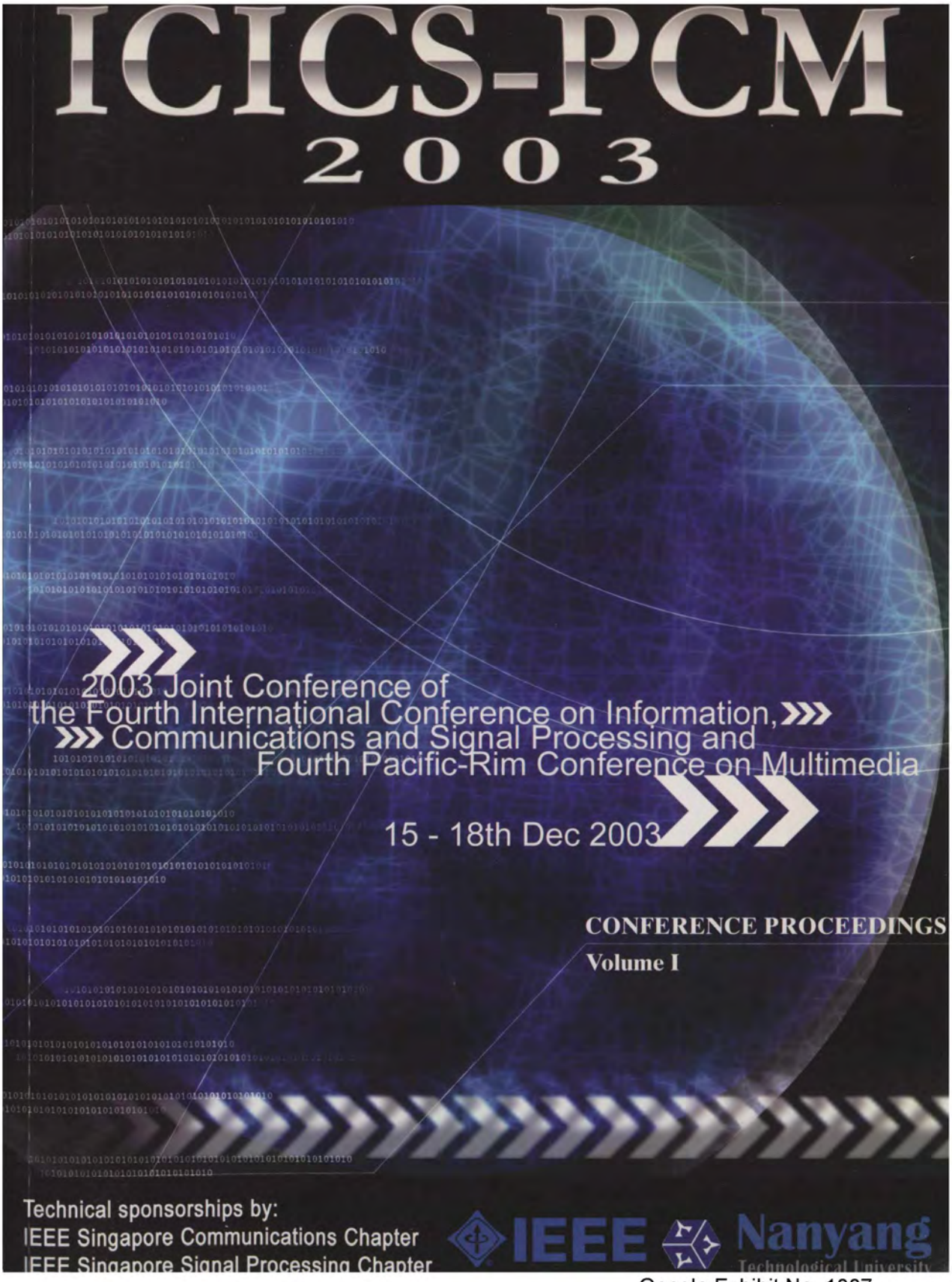
[9] V. Lappalainen, *et al*, "Performance of H.26L video encoder on general purpose processor," *Journal of VLSI Signal Processing*, pp. 239-249, 2003.

[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.

Google Exhibit No. 1007
Page 005

Google Exhibit No. 1063
Page 0492

ICICS-PCM 2003



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

15 - 18th Dec 2003

CONFERENCE PROCEEDINGS

Volume I

Technical sponsorships by:
IEEE Singapore Communications Chapter
IEEE Singapore Signal Processing Chapter

IEEE

Nanyang Technological University

Google Exhibit No. 1007
Page 006

Google Exhibit No. 1063
Page 0493